

John -

This is the ISPF/PDF
Manual with a description
of Super C Utility
Also enclosed is a manual
for another comparison utility,
Comparex. If you have
questions, let me know.
I can get in touch w/
some techies who are pretty
good at translating this stuff.

Seth



Technical Newsletter

This Newsletter No.	SN28-4654-0
Date	April 18, 1988
Base Publication No.	SC34-4118-0
File No.	S370/4300-39
Previous Newsletters	None

ISPF/PDF Version 2 Release 3

Guide for MVS

© Copyright IBM Corporation 1984, 1985, 1987, 1988

This Technical Newsletter, a part of the Interactive System Productivity Facility/Program Development Facility (ISPF/PDF) Guide, Version 2 Release 3 for MVS, provides replacement and addendum pages for that publication. This material remains in effect for subsequent releases unless specifically altered.

Pages to be replaced are:

xix through xxxiv
103 through 114
137 through 140
341 through 342
413 through 438

Pages to be added are:

114.1 through 114.8
140.1 through 140.2
342.1 through 342.2

Technical changes are indicated by an asterisk (*) to the left of the changes.

NOTE: Please file the cover letter in your publication to provide a record of changes.

Job Statement Information	55
Running a Sample PDF Session	56
 Chapter 3. ISPF Parms (Option 0)	63
Terminal (Option 0.1)	64
Log/List (Option 0.2)	67
Log and List Data Set Default Options	68
Changing Primary and Secondary Page Values	69
PF Keys (Option 0.3)	70
Terminals with 12 PF Keys	70
Terminals with 24 PF Keys	70
Assigning ISPF/PDF Commands	70
Assigning Labels	72
Display (Option 0.4)	74
List (Option 0.5)	76
Printing Panels with Long Lines	77
Graphic (Option 0.6)	78
Environ (Option 0.7)	81
Using the DUMP Parameter	84
 Chapter 4. Browse (Option 1)	85
Browsing a Data Set	85
Browse Data Display	86
Ending Browse	87
Browse Primary Commands	87
BROWSE - Browsing Recursively	87
COLUMNS - Identifying Columns	88
DISPLAY - Controlling the Display	89
FIND - Finding Character Strings	90
Specifying Find Strings	90
Omitting String Delimiters	91
Using String Delimiters	91
Starting Point, Direction, and Extent of Search	91
Conditions for Character String Matches	92
Use of Text Strings	92
Use of Character Strings	93
Use of Picture Strings	93
Column Limitations	94
Using RFINd	94
HEX - Displaying Data in Hexadecimal Format	95
LOCATE - Locating Lines	96
RESET - Removing the Column-Identification Line	97
 Chapter 5. Edit (Option 2)	99
Editing a Data Set	99
 Chapter 6. Utilities (Option 3)	101
Library Utility (Option 3.1)	102
Blank - Display Member List	102
B - Browse Member	103
C - Compress Data Set	103
P - Print Member	103
X - Print Index Listing	103
R - Rename Member	104

L - Print Entire Data Set	104
D - Delete Member	104
I - Data Set Information	104
Information for Managed Data Sets	105
S - Data Set Information (Short)	106
Information (Short) for Managed Data Sets	107
Data Set Utility (Option 3.2)	108
A - Allocate New Data Set	108
C - Catalog Data Set	114
Cataloging Managed Data Sets	114
R - Rename Entire Data Set	114
Renaming GDG Data Sets	114.1
Renaming Managed Data Sets	114.1
U - Uncatalog Data Set	114.1
Uncataloging Managed Data Sets	114.1
D - Delete Entire Data Set	114.1
S - Data Set Information (Short)	114.3
Blank - Data Set Information	114.3
M - Enhanced Data Set Allocation	114.3
Allocation Errors	114.7
Move/Copy Utility (Option 3.3)	115
C and CP - Copying Data Sets	116
C - Copy Data Set or Member(s)	116
CP - Copy and Print	116
M and MP - Moving Data Sets	118
M - Move Data Set or Member(s)	118
MP - Move and Print	118
L and LP - Copying and Locking Data Sets	119
L - Copy and Lock Member(s)	119
LP - Copy, Lock, and Print	119
P and PP - Promoting Data Sets	120
P - Promote Data Set or Member(s)	120
PP - Promote and Print	120
Using the Move/Copy Utility without Load Modules	122
Using the Move/Copy Utility with Load Modules	122
Moving, Copying, or Promoting Alias Entries	123
Data Set List Utility (Option 3.4)	124
Blank - Display Data Set List	127
Line Commands	131
TSO Commands and CLISTs	132
Using the Slash (/) Symbol	132
TSO Command/CLIST Variables	134
P - Print Data Set List	135
V - Display VTOC Information	136
PV - Print VTOC Information	137
Data Set List Utility Line Commands	138
B - Browse Data Set	138
E - Edit Data Set	138
D - Delete Data Set	139
R - Rename Data Set	139
I - Data Set Information	139
S - Information (Short)	139
C - Catalog Data Set	140
U - Uncatalog Data Set	140

P - Print Data Set	140
X - Print Index Listing	140
M - Display Member List	140.1
Z - Compress Data Set	141
F - Free Unused Space	141
= - Repeat Last Command	141
Data Set List Utility Primary Commands	142
CONFIRM Command	142
FIND and RFIND Commands	142
LOCATE Command	143
SAVE Command	143
SHOWCMD Command	144
SORT Command	146
Reset ISPF Statistics Utility (Option 3.5)	147
R - Reset (Create/Update) ISPF Statistics	148
D - Delete ISPF Statistics	149
Results of Resetting Statistics	149
Hardcopy Utility (Option 3.6)	150
Generating and Submitting JCL	151
Additional Batch Printing Information	153
Using the TSO/E Information Center Facility	153
Removal of List VTOC Utility (Option 3.7)	155
Outlist Utility (Option 3.8)	156
L - List Job Names/IDs Via the TSO STATUS Command	157
D - Delete Job Output from SYSOUT Hold Queue	157
P - Print Job Output and Delete from SYSOUT Hold Queue	157
R - Requeue Job Output to a New Output Class	158
Blank - Display Job Output	158
Command Table Utility (Option 3.9)	159
Scrolling a Command Table	162
Saving a Command Table	162
Cancelling a Command Table	162
Command Table Line Commands	162
D - Deleting Lines	162
I - Inserting Lines	162
R - Repeating Lines	162
Convert Menus/Messages (Option 3.10)	163
1 - Convert Old Format Menus to New Format Panels	164
2 - Convert Old Format Messages to New Format Messages	165
Format Specification Utility (Option 3.11)	166
A - Add a New Format	167
C - Copy Formats	168
D - Delete a Format	169
U - Update a Format	169
L or BLANK - Display Format List	169
Format Selection List Commands	170
Deleting a Format	170
Locating Format Names	170
Renaming a Format	171
Sorting Format Names	171
Updating or Selecting a Format	171
SuperC Utility (Option 3.12)	172
SuperC Member Lists	176
Submitting a SuperC Job in Batch Mode	177

Blank - Generate Output Listing to SYSOUT CLASS	179
1 - Generate Output Listing in DATA SET NAME	179
Specifying Existing Data Sets	179
Specifying Nonexistent Data Sets	180
2 - Generate Output Listing Using Completed //OUTDD DD	180
Using the NOLIST Listing Type in Batch Mode	181
SuperCE Utility (Option 3.13)	182
Blank - Compare Data Sets	186
B - Submit Batch Data Set Compare	187
Printing a SuperCE Listing in Batch Mode	188
S - Extended Search-For Compare Utility	188
Blank - Search-For Strings	191
P - Select Search-For Process Options	192
E - Edit Search-For Statements Data Set	193
P - Select SuperCE Process Options	194
E - Edit SuperCE Statements Data Set	194
A - Activate/Create Profiles	195
A - Activate	197
C - Create	197
Search-For Utility (Option 3.14)	199
Entering Multiple Search Strings	200
Search-For Strings and Keywords	201
Entering Search Strings	201
Using Keywords	201
Search-For Member Lists	202
 Chapter 7. Foreground (Option 4)	205
Foreground Processing Sequence	205
Expanding Packed Data	211
Restrictions on Member Expansion and Member Parts Lists	213
Member Expansion Return Codes	215
Trigger Statement Errors	215
Input Data Sets	215
List Data Sets	216
Password Protection	217
Object Data Sets	217
Foreground - TSO/E Information Center Facility	218
Assembler H (Option 4.1)	219
Assembler XF (Option 4.1A)	220
VS COBOL II Compiler (Option 4.2)	221
OS/VS COBOL Compiler (Option 4.2A)	223
VS FORTRAN Compiler (Option 4.3)	225
PL/I Checkout Compiler (Option 4.4)	226
Interpretable Text Data Sets	227
PL/I Optimizing Compiler (Option 4.5)	228
Pascal/VS Compiler (Option 4.6)	229
Linkage Editor (Option 4.7)	231
Linkage Editor Concatenation Sequence	232
SCRIPT/VS (Option 4.9)	233
SCRIPT/VS Processing Sequence	233
Selecting a Formatting Style	237
Changing Style Options	238
Using SCRIPT/VS with the TSO/E Information Center Facility	239
VS COBOL II Interactive Debug (Option 4.10)	240

COBOL Interactive Debug (Option 4.10A)	241
COBOL Interactive Debug Processing Sequence	241
Symbolic Debug Data Sets	242
Print Output Data Sets	243
FORTRAN Interactive Debug (Option 4.11)	244
Member Parts List (Option 4.12)	245
Member Not Found	249
 Chapter 8. Batch (Option 5)	251
Batch Processing Sequence	251
JCL Generation - Compilers	256
JCL Generation - Assemblers and Linkage Editor	257
Assembler H (Option 5.1)	258
Assembler XF (Option 5.1A)	260
VS COBOL II Compiler (Option 5.2)	262
OS/VS COBOL Compiler (Option 5.2A)	264
VS FORTRAN Compiler (Option 5.3)	265
PL/I Checkout Compiler (Option 5.4)	267
PL/I Optimizing Compiler (Option 5.5)	268
Pascal/VS Compiler (Option 5.6)	269
Linkage Editor (Option 5.7)	270
VS COBOL II Interactive Debug (Option 5.10)	272
Member Parts List (Option 5.12)	273
 Chapter 9. Command (Option 6)	275
Entering TSO Commands and CLISTs	276
Rules for Entering TSO Commands	276
Rules for Entering CLISTs	277
Using the Session Manager	277
 Chapter 10. Dialog Test (Option 7)	279
Environment	280
Variable Usage	281
Severe Error Handling	282
Commands	282
Ending the Current Option Without Saving Changes	284
Saving Changes	284
Finding a Character String	284
Displaying Breakpoint Qualification Data	285
Restoring the Format of the Breakpoints Panel	285
Dialog Test Line Commands	285
D - Deleting Lines	285
I - Inserting Lines	285
R - Repeating Lines	286
Functions (Option 7.1)	287
Panels (Option 7.2)	291
Variables (Option 7.3)	293
Variables Commands	294
Manipulating Variables	294
Creating New Variables	295
Deleting Variables	295
Variables Usage Notes	295
Tables (Option 7.4)	298
1 - Display Row	300

Display Row Commands	301
2 - Delete Row	301
3 - Modify Row	301
Modify Row Commands	303
4 - Add Row	304
Add Row Commands	305
5 - Display Structure	306
Display Structure Command	306
6 - Display Status	307
Table Not Open	307
Table Open	308
Log (Option 7.5)	310
ISPF Log Not Available	310
Trace Output in ISPF Log	310
Trace Header Entries	310
Function Trace Entries	311
Variable Trace Entries	311
Dialog Services (Option 7.6)	313
Traces (Option 7.7)	315
1 - Function Traces	315
Function Traces Commands	317
2 - Variable Traces	317
Variable Traces Commands	318
Breakpoints (Option 7.8)	319
Specifying Breakpoints	319
Breakpoints Commands	320
Qualification Parameter Values	320
Finding a Breakpoint	323
Tutorial (Option 7.T)	326
Exit (Option 7.X)	326
 Chapter 11. LM Utilities (Option 8)	 327
 Chapter 12. IBM Products (Option 9)	 329
CSP/AD (Option 9.1)	330
CSP/AE (Option 9.2)	331
INFO/SYS (Option 9.3)	332
COBOL/SF-F (Option 9.4)	333
COBOL/SF-B (Option 9.5)	334
 Chapter 13. Tutorial (Option T)	 335
Selecting the Tutorial Option (T)	335
Selecting a Tutorial Topic	335
Ending the Tutorial	337
Using the HELP Command	338
 Appendix A. Listing Formats	 339
Source and Index Listings	339
Source Listings	340
Index Listings	341
Index Listings for Source Libraries	341
Index Listings for Managed Source Libraries	342
Index Listings for Load Libraries	342.1
Index Listings for Managed Load Libraries	342.2

ISPF Log Listings	343
Member List Listings	344
Member List Listings for Source Libraries	344
Member List Listings for Load Libraries	345
Formats for Member List Listings	346
Data Set List Listings	347
SuperC Listings	348
SuperC Listing Title Lines	349
Listing Output Section	352
Listing Prefix Codes	352
Type of Difference Codes	353
Member Summary Listing	355
Summary and Statistics/PDS Overall Totals	355
Bottom-of-Listing Information	356
Side-By-Side Listings	357
Effect of FILE Compare Type on SuperC Listings	358
FILE Comparison of a Sequential Data Set or Membered PDS ..	358
FILE Comparison of a Complete PDS	359
Search-For Listings	360
 Appendix B. Character Translations for APL and TEXT	
Keyboards	363
 Appendix C. List of Abbreviations	365
Command Name Abbreviations	365
Field Value Abbreviations	365
Keyword/Operand Abbreviations	366
Scroll Amount Abbreviations	366
Programming Language Abbreviations	366
 Appendix D. Allocation Data Sets	367
SYSIN Data Set	367
SYSLIB Data Set	367
SYSPRINT Data Set	368
SYSTEM Data Set	368
SYSLIN Data Set	368
SYSPUNCH Data Set	369
SYSUT1 Data Set	369
SYSUT2 - SYSUT5 Data Sets	369
 Appendix E. SuperC Reference	371
Utility Differences	371
Program Description	372
Comparison Result Differences	374
Applications	375
Return Codes	377
Process Options	378
Input Process Control Options	379
"Don't Process" Control Options	380
Output Process Control Options	381
Update Data Set Control Options	382
APNDUPD - Append Update Data Set	383
UPDCMS8 - Update CMS Sequenced 8	383
UPDCNTL - Update Control	384

UPDCNTL Data Set Using LINE Compare	384
UPDCNTL Data Set Using WORD Compare	385
UPDCNTL Data Set Using BYTE Compare	386
UPDMVS8 - Update MVS Sequenced 8	387
UPDPDEL - Prefixed Delta Lines	388
UPDSEQ0 - Update Sequenced 0	389
UPDSUMO - Update Summary Only	389
UPDSUMO Data Set Using LINE Compare	390
UPDSUMO Data Set Using WORD Compare	391
UPDSUMO Data Set Using BYTE Compare	392
Process Statements	393
Process Statement Notation Conventions	393
CMPBOFS - Compare Byte Offsets	395
Compare Format	395
Parameters	395
Example	395
CMPCOLM - Compare Columns	396
Compare and Search Format	396
Parameters	396
Example	396
CMPCOLMN/CMPCOLMO - Compare Column New and Old	397
Compare Format	397
Parameters	397
Example	398
CMPLINE - Compare Line	399
Compare and Search Format	399
Parameters	399
Examples	400
Comment Lines	401
Compare and Search Format	401
DPLINE/DPLINEC - Don't Process This Line	402
Compare and Search Format	402
Parameters	402
Examples	403
LNCT - Line Count	404
Compare and Search Format	404
Parameters	404
Example	404
LSTCOLM - List Columns	405
Compare and Search Format	405
Parameters	405
Example	405
NCHGT/OCHGT - New and Old Change Text	406
Compare Format	406
Search Format	406
Parameters	406
Examples	407
NTITLE/OTITLE - Alternate Title for Input Data Sets	408
Compare Format	408
Search Format	408
Parameter	408
Example	408
SELECT - Select PDS Members	409
Compare Format	409

Search Format	409
Compare Parameters	409
Search Parameter	409
Examples	409
Compare Example	409
Search Example	410
SLIST - Include Statements in Listing	411
Compare and Search Format	411
Parameters	411
Example	411
SRCHFOR/SRCHFORC - Search-For String	412
Search Format	412
Parameters	412
Examples	412
Index	413

Figures

1.	ISPF/PDF Primary Option Menu	5
2.	Panel Format	7
3.	Command Types and Entry Methods	10
4.	ISPF Commands Assigned to PF Keys	12
5.	Other ISPF Commands	13
6.	Line Command Field Characteristics	16
7.	Default PF Key Settings	20
8.	Split Screen Example	21
9.	Specify Disposition of Log and List Data Sets Panel	23
10.	Hierarchy of ISPF Libraries	33
11.	Member Selection List Differences	38
12.	Member List Display	40
13.	Load Module Library Display	41
14.	SORT Fields for Source Libraries	46
15.	SORT Fields for Load Libraries	46
16.	Library Utility - Before and After Print, Rename, and Delete	48
17.	ISPF Parameter Options Panel	63
18.	Terminal Characteristics Panel	64
19.	Log and List Defaults Panel	67
20.	PF Key Definitions and Labels Panel (12 PF Keys)	70
21.	PF Key Definitions and Labels Panels (24 PF Keys)	71
22.	Sample Screen with PF Key Definition Lines	73
23.	Display Characteristics Panel	74
24.	List Data Set Characteristics Panel	76
25.	Print Graphics Parameters Panel	78
26.	Screen Containing Graphics to be Printed	79
27.	Example of Using Aspect Ratio Parameter 0	79
28.	Example of Using Aspect Ratio Parameter 1	80
29.	ISPF ENVIRON Command Settings Panel	81
30.	Browse - Entry Panel	85
31.	Browse - Data Display	86
32.	Browse - Column-Identification Line	89
33.	Browse Hexadecimal Display - Vertical	96
34.	Browse Hexadecimal Display - Data	96
35.	Edit - Entry Panel	99
36.	Utility Selection Menu	101
37.	Library Utility Panel	102
38.	Library Utility - Data Set Information	104
* 38.1.	Library Utility - Information for Managed Data Sets	105
* 38.2.	Library Utility - Information for Managed Data Sets with No	
* Directory Block Information		106
39.	Library Utility - Data Set Information (Short)	106
* 39.1.	Library Utility - Information (Short) for Managed Data Sets	107
40.	Data Set Utility Panel	108
41.	Allocate New Data Set Panel - Two-digit Year Support	109
42.	Allocate New Data Set Panel - Four-digit Year Support	113
43.	Rename Data Set Panel	114

	44. Confirm Delete Panel	114.2
	45. Confirm Purge Panel	114.2
*	45.1. Allocate New Data Set Panel - Managed Data Set Support ...	114.4
*	45.2. Allocation Error Panel	114.7
	46. Move/Copy Utility Panel	115
	47. Move/Copy Utility - "To" Panel for Copying	116
	48. Move/Copy Utility - "To" Panel for Moving	118
	49. Move/Copy Utility - "Target" Panel for Promoting	121
	50. Data Set List Utility Panel	124
	51. Sequence of Data Set List Display Views	126
	52. Data Set List - VOLUME View	128
	53. Data Set List - SPACE View	128
	54. Data Set List - ATTRIB View	129
	55. Data Set List - TOTAL View	129
	56. Data Set List Utility Line Commands and Messages	130
	57. TSO Command/CLIST Variables (Output)	134
	58. TSO Command/CLIST Variables (Input)	135
	59. VTOC Information Display	136
	60. Member List Display - Expanded Line Command Field	140.1
	61. Load Module Library Display - Expanded Line Command Field ..	141
	62. Data Set List Utility - SHOWCMD Panel	145
	63. SORT Fields for Source Libraries	146
	64. Reset ISPF Statistics Panel	147
	65. Reset Statistics Field Combinations and Results	149
	66. Hardcopy Utility Panel - Before JCL Generation	150
	67. Hardcopy Utility Panel - After JCL Generation	152
	68. Hardcopy Utility Panel - With the TSO/E Information Center Facility Installed	154
	69. Outlist Utility Panel	156
	70. Command Table Utility Panel	159
	71. Command Table Editing Panel	160
	72. Convert Menus/Messages Panel	163
	73. Copy Panel for Specifying New Format Data Set	164
	74. Format Specification Panel	166
	75. Format Definition Panel	167
	76. Copy Format Selection List Panel	169
	77. Format Selection List Panel	170
	78. SuperC Utility Panel	172
	79. SuperC Utility - "Old" Data Set Panel	173
	80. SuperC Member List Panel	177
	81. SuperC Utility - Submit Batch Jobs Panel	178
	82. SuperC Utility - Submit Batch Jobs Panel using NOLIST	181
	83. SuperCE Utility Panel	182
	84. SuperCE - Concatenation Interactive Entry Panel	187
	85. SuperCE - Concatenation Batch Entry Panel	188
	86. Extended Search-For Utility Panel	189
	87. Extended Search-For - Concatenation Data Set Entry Panel	192
	88. Search-For Process Statements Panel	193
	89. SuperC Process Statements Panel	195
	90. SuperCE Profile Browse Display	196
	91. SuperCE - Profile Manager Panel	196
	92. Search-For Utility Panel	199
	93. Multiple Search Strings Panel	201
	94. Search Member List Panel	203
	Foreground Selection Panel	205

96.	Foreground Print Options Panel	210
97.	Expansion Triggers and Syntax	212
98.	Foreground Print Options Panel With TSO/E Information Center Facility	218
99.	Foreground Assembler H Panel	219
100.	Foreground Assembler H Options	219
101.	Foreground Assembler XF Panel	220
102.	Foreground Assembler XF Options	220
103.	Foreground VS COBOL II Compile Panel	221
104.	Foreground VS COBOL II Compiler Options	222
105.	Foreground OS/VS COBOL Compile Panel	223
106.	Foreground OS/VS COBOL Compiler Options	224
107.	Foreground VS FORTRAN Compile Panel	225
108.	Foreground VS FORTRAN Compiler Options	225
109.	Foreground PL/I Checkout Compile Panel	226
110.	Foreground PL/I Checkout Compiler Options	227
111.	Foreground PL/I Optimizing Compile Panel	228
112.	Foreground PL/I Optimizing Compiler Options	228
113.	Foreground Pascal/VS Compile Panel	229
114.	Foreground Pascal/VS Compiler Options	229
115.	Foreground Linkage Edit Panel	231
116.	Foreground Linkage Editor Options	231
117.	SCRIPT/VS Processor Panel	233
118.	Foreground Print Options for Style Panel	236
119.	Select SCRIPT/VS Formatting Style Panel	237
120.	SCRIPT/VS Options for Style Panel: Draft	238
121.	Foreground Print Options for Style Panel With the TSO/E Information Center Facility	239
122.	Foreground VS COBOL II Interactive Debug Panel	240
123.	COBOL Interactive Debug Panel	241
124.	FORTRAN Interactive Debug Panel	244
125.	Foreground FORTRAN Interactive Debug Options	244
126.	Foreground Member Parts List Panel	245
127.	Member Parts List Display	247
128.	Foreground Member Parts List Record Formats	248
129.	Batch Selection Panel	251
130.	Batch Selection Panel With JCL Generated	255
131.	ISRSCAN and ISRLEMX Return Codes	257
132.	Batch Assembler H Panel	258
133.	Batch Assembler H Options	259
134.	Batch Assembler XF Panel	260
135.	Batch Assembler XF Options	261
136.	Batch VS COBOL II Compile Panel	262
137.	Batch VS COBOL II Compiler Options	263
138.	Batch OS/VS COBOL Compile Panel	264
139.	Batch OS/VS COBOL Compiler Options	264
140.	Batch VS FORTRAN Compile Panel	265
141.	Batch VS FORTRAN Compiler Options	266
142.	Batch PL/I Checkout Compile Panel	267
143.	Batch PL/I Checkout Compiler Options	267
144.	Batch PL/I Optimizing Compile Panel	268
145.	Batch PL/I Optimizing Compiler Options	268
146.	Batch Pascal/VS Compile Panel	269
147.	Batch Pascal/VS Compiler Options	269
148.	Batch Linkage Edit Panel	270

149.	Batch Linkage Editor Options	271
150.	VS COBOL II Interactive Debug Panel	272
151.	Batch Member Parts List Panel	273
152.	TSO Command Processor Panel	275
153.	Dialog Test Primary Option Menu	280
154.	Invoke Dialog Function/Selection Menu	287
155.	Display Panel	291
156.	Message Display Panel	292
157.	Variables Panel	293
158.	Tables Panel	298
159.	Display Row Panel	300
160.	Confirm Table Row Delete Panel	301
161.	Modify Row Panel	302
162.	Add Row Panel	304
163.	Table Structure Panel	306
164.	Status for Table Panel with Table Not Open	307
165.	Status for Table Panel with Table Open	308
166.	Invoke Dialog Service Panel	313
167.	Special Display Panel	314
168.	Traces Panel	315
169.	Function Traces Panel	316
170.	Variable Traces Panel	317
171.	Breakpoints Panel	319
172.	Breakpoints Panel with Qualification Parameter Values	321
173.	Breakpoint Primary Option Menu	323
174.	Dialog Test Tutorial - First Panel	326
175.	Library Management Utilities Panel	327
176.	Additional IBM Program Development Products Panel	329
177.	Cross System Product/Application Development Panel	330
178.	Cross System Product/Application Execution Panel	331
179.	Information/System Panel	332
180.	COBOL Structuring Facility Panel - Foreground Dialog	333
181.	COBOL Structuring Facility Panel - Background Dialog	334
182.	ISPF Tutorial and Introduction Panels	336
183.	Table of Contents Panel	337
184.	Help Tutorial Panel	338
185.	Sample Source Listing	340
186.	Sample Index Listing - Source Library	341
*	186.1. Sample Index Listing - Managed Source Library	342
	187. Sample Index Listing - Load Library	342.1
*	187.1. Sample Index Listing - Managed Load Library	342.2
	188. Sample ISPF Log Listing	343
	189. Sample Member List Listing for a Source Library	344
	190. Sample Member List Listing for a Load Library	345
	191. Format of Source Library Member List Listing	346
	192. Format of Source Library Member List Listing	346
	193. Sample Data Set List Listing	347
	194. Format of Data Set List Listing	347
	195. Split Screen Showing Source Data for SuperC Listings	349
	196. DELTA Listing of a LINE Comparison of Two PDSs	350
	197. LONG Listing of a WORD Comparison of Two PDSs	351
	198. WORD Comparison of Two PDSs Using Change Bar to Show Differences	354
	199. Side-By-Side LINE Comparison Using the NARROW Process Option	357

200.	FILE Comparison of a Membered PDS	358
201.	FILE Comparison of a Complete PDS	359
202.	Search-For Listing	360
203.	Internal Character Representations for APL Keyboards	363
204.	Internal Character Representations for TEXT Keyboards	364
205.	Input Process Control Options	379
206.	"Don't Process" Control Options	380
207.	Output Process Control Options	381
208.	UPDCMS8 Update Data Set	383
209.	UPDCNTL Data Set Using LINE Compare	384
210.	UPDCNTL Format Using LINE Compare	384
211.	UPDCNTL Listing Using WORD Compare	385
212.	UPDCNTL Format Using WORD Compare	385
213.	UPDCNTL Listing Using BYTE Compare	386
214.	UPDCNTL Format Using BYTE Compare	387
215.	UPDMVS8 Update Data Set Listing	387
216.	UPDPDEL Update Data Set Listing	388
217.	UPDSEQ0 Update Data Set	389
218.	UPDSUMO Data Set Using LINE Compare	390
219.	UPDSUMO Format Using LINE Compare	390
220.	UPDSUMO Data Set Using WORD Compare	391
221.	UPDSUMO Format Using WORD Compare	391
222.	UPDSUMO Data Set Using BYTE Compare	392
223.	UPDSUMO Format Using BYTE Compare	392

Library Utility (Option 3.1)

B - Browse Member

If you select option B, you must specify a partitioned data set and a member name. The specified member is displayed in Browse mode. You can use all the Browse commands. When you end Browse by entering the END command, the Library Utility panel is displayed again.

C - Compress Data Set

If you select option C, you can specify any partitioned data set. The compress is accomplished by invoking either of the following:

- The IEBCOPY utility
- An optional compress request exit routine, which can be specified by your installation.

P - Print Member

If you select option P, you must specify a partitioned data set and a member name. A source listing of the member is recorded in the ISPF list data set.

Note: If any members are to be printed, the data set characteristics must conform to those for the L option.

X - Print Index Listing

If you select option X, you must specify either a DASD-resident sequential or partitioned data set. The index listing is recorded in the ISPF list data set. For a partitioned data set, the index listing includes general information about the data set followed by a member list. For a sequential data set, the index listing includes general information only. See:

- * • Figure 186 on page 341 and Figure 187 on page 342.1 for examples of the
* regular index listing format for source libraries and load libraries,
* respectively. In these types of listings, the 1st extent quantity,
* secondary quantity, current allocation, and current utilization sizes are
* shown in tracks for data sets that are allocated in bytes, kilobytes, or
* megabytes on a volume that is *not* managed by the Storage Management
* Subsystem.
- * • Figure 186.1 on page 342 and Figure 187.1 on page 342.2 for examples of
* the index listing format for source libraries and load libraries,
* respectively, on Storage Management Subsystem volumes. In these
* index listings, the management, storage, and data classes used are
* shown under the GENERAL DATA heading. Also, the 1st extent
* quantity, secondary quantity, current allocation, and current utilization
* sizes can be shown in bytes, kilobytes, or megabytes, in addition to
* tracks, blocks, or cylinders. See "M - Enhanced Data Set Allocation"
* on page 114.3 if you need more information.

Note: If PDF was entered in TEST mode, the listing also includes TTR data for each member of the data set. This data is the track and record address, where the members reside on the volume.

R - Rename Member

If you select option R, you must specify a partitioned data set and member name. You must also specify a new member name in the NEWNAME field.

L - Print Entire Data Set

If you select option L, you must specify either a DASD-resident sequential or partitioned data set. The allowable data set characteristics are the same as for Browse, except that data sets with a logical record length greater than 300 characters are not printed. Also, the data should not contain any printer control characters. Use the Hardcopy utility (3.6) to print data sets that contain printer control characters. A source listing of the complete data set (including all members of a partitioned data set), preceded by an index listing, is recorded in the ISPF list data set.

D - Delete Member

If you select option D, you must specify a partitioned data set and member name.

I - Data Set Information

If you select option I, the location, characteristics, and current space utilization of the specified data set are displayed (Figure 38). For sequential data sets, options I and S display the same information.

```

----- DATA SET INFORMATION -----
COMMAND ----> _

DATA SET NAME: ISPFDEMO.MYLIB.PLI

GENERAL DATA:                                CURRENT ALLOCATION:
Volume serial:      D00163                      Allocated blocks:      3
Device type:        3350                        Allocated extents:     1
Organization:       PO                          Maximum dir. blocks:   30
Record format:      VB
Record length:      255
Block size:         6160
1st extent blocks:  3
Secondary blocks:    5

Creation date:      1986/10/01
Expiration date:    ***NONE***

CURRENT UTILIZATION:
Used blocks:        3
Used extents:       1
Used dir. blocks:   1
Number of members:  1
    
```

Figure 38. Library Utility - Data Set Information

Library Utility (Option 3.1)

The Allocated blocks and Used blocks fields can vary, depending on the value that was specified in the SPACE UNITS field when you allocated the data set. For example, the preceding panel shows what the Data Set Information panel would look like if the data set was allocated by specifying BLOCKS in the SPACE UNITS field.

The 1st extent quantity, secondary quantity, current allocation, and current utilization sizes are shown in tracks for data sets that are allocated in bytes, kilobytes, or megabytes on a volume that is *not* managed by the Storage Management Subsystem. See "Information for Managed Data Sets" if you need more information.

* Information for Managed Data Sets

Figure 38.1 shows the information that is displayed for data sets that reside on Storage Management Subsystem volumes when the directory block information is available.

```

----- DATA SET INFORMATION -----
COMMAND ===> _
DATA SET NAME: ISPFDEMO.MYLIB.PLI

GENERAL DATA:                                CURRENT ALLOCATION:
Management Class:      M1                      Allocated kilobytes:      3
Storage Class:         S1                      Allocated extents:       1
Volume serial:         D00163                  Maximum dir. blocks:    30
Device type:           3350

Data Class:            D1                      CURRENT UTILIZATION:
Organization:          PO                      Used kilobytes:          3
Record format:         VB                      Used extents:            1
Record length:         255                    Used dir. blocks:        1
Block size:            6160                    Number of members:       1
1st extent kilobytes:   3
Secondary kilobytes:    5

Creation date:          1987/12/02
Expiration date:        ***NONE***

```

Figure 38.1. Library Utility - Information for Managed Data Sets

The major difference between this information and the information that is displayed for data sets on non-managed volumes is the addition of the following classes:

- Management class
- Storage class
- Data class.

Also, the 1st extent quantity, secondary quantity, current allocation, and current utilization sizes can be shown in bytes, kilobytes, or megabytes, in addition to tracks, blocks, or cylinders. See "M - Enhanced Data Set Allocation" on page 114.3 for information about allocating data sets using the classes listed above.

Figure 38.2 on page 106 shows the information that is displayed for managed data sets when the directory block information is *not* available.

Library Utility (Option 3.1)

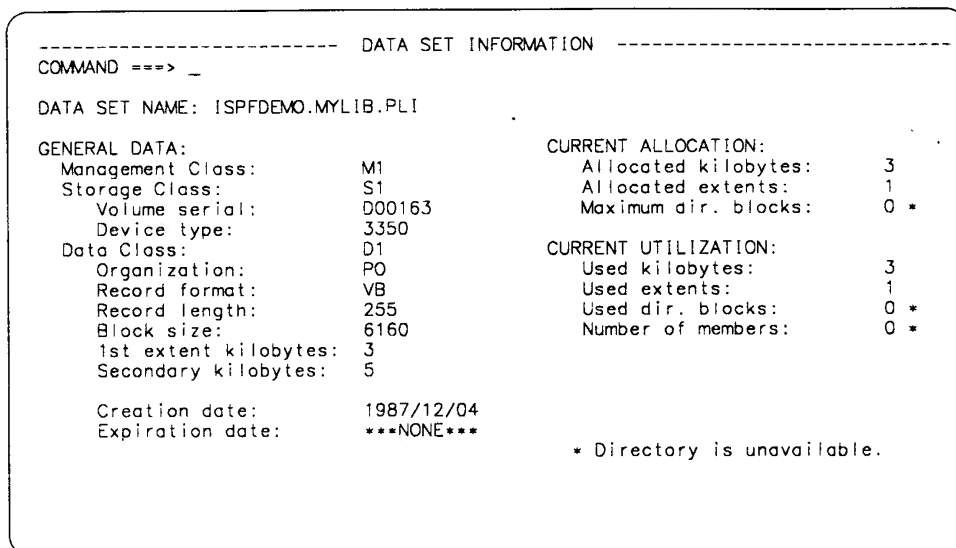


Figure 38.2. Library Utility - Information for Managed Data Sets with No Directory Block Information

S - Data Set Information (Short)

If you select option S, information about the selected data set is displayed. The information displayed by option S is the same information displayed by option S of the Data Set utility (3.2), but it differs from option I in two respects.. Information for partitioned data sets, when displayed by option S, lacks the number of maximum and used directory blocks, and the number of members. For sequential data sets, options I and S display the same information. Figure 39 shows a short format example of data set information for a partitioned data set.

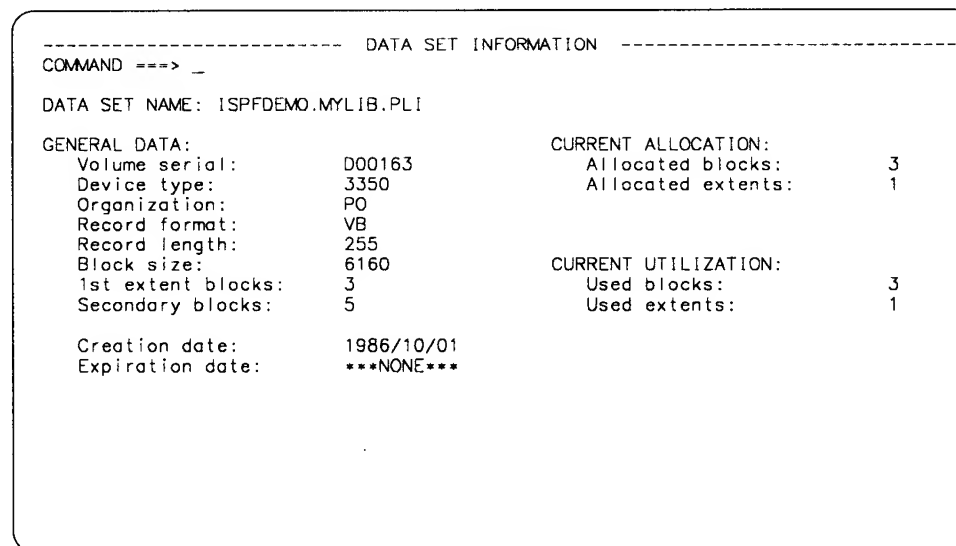


Figure 39. Library Utility - Data Set Information (Short)

Library Utility (Option 3.1)

The Allocated blocks and Used blocks fields can vary, depending on the value that was specified in the SPACE UNITS field when you allocated the data set. For example, the preceding panel shows what the short format of the Data Set Information panel would look like if the data set was allocated by specifying BLOCKS in the SPACE UNITS field.

The 1st extent quantity, secondary quantity, current allocation, and current utilization sizes are shown in tracks for data sets that are allocated in bytes, kilobytes, or megabytes on a volume that is *not* managed by the Storage Management Subsystem. See “Information (Short) for Managed Data Sets” if you need more information.

* Information (Short) for Managed Data Sets

Figure 39.1 shows the short form of the information that is displayed for data sets that reside on Storage Management Subsystem volumes.

```

----- DATA SET INFORMATION -----
COMMAND ==> _

DATA SET NAME: ISPFDEMO.MYLIB.PLI

GENERAL DATA:                                CURRENT ALLOCATION:
Management Class:      M1                      Allocated kilobytes:      3
Storage Class:         S1                      Allocated extents:       1
Volume serial:         D00163
Device type:           3350
Data Class:            D1
Organization:          PO
Record format:         VB
Record length:         255
Block size:            6160
1st extent kilobytes:  3
Secondary kilobytes:   5

                                CURRENT UTILIZATION:
                                Used kilobytes:      3
                                Used extents:         1

Creation date:          1987/12/02
Expiration date:        ***NONE***
  
```

Figure 39.1. Library Utility - Information (Short) for Managed Data Sets

The major difference between this information and the information that is displayed for data sets on non-managed volumes is the addition of the following classes:

- Management class
- Storage class
- Data class.

Also, the 1st extent quantity, secondary quantity, current allocation, and current utilization sizes can be shown in bytes, kilobytes, or megabytes, in addition to tracks, blocks, or cylinders. See “M - Enhanced Data Set Allocation” on page 114.3 for information about allocating data sets using the classes listed above.

Data Set Utility (Option 3.2)

Data Set Utility (Option 3.2)

When you select this option, a panel is displayed that allows you to specify a data set and show an action to be done (Figure 40).

```

----- DATA SET UTILITY -----
OPTION ===> _

A - Allocate new data set          C - Catalog data set
R - Rename entire data set        U - Uncatalog data set
D - Delete entire data set        S - Data set information (short)
blank - Data set information      M - Enhanced data set allocation

ISPF LIBRARY:
PROJECT ===> ISPFDEMO
GROUP   ===> MYLIB
TYPE    ===> PLI

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ===>
VOLUME SERIAL ===>          (If not cataloged, required for option "C")

DATA SET PASSWORD ===>      (If password protected)

```

Figure 40. Data Set Utility Panel

Data Facility Product (DFP) Version 3 Release 1 must be installed and available for option M to appear on this panel. Also, the Storage Management Subsystem must be active. See “M - Enhanced Data Set Allocation” on page 114.3 for information about managed data sets.

All the fields on the Data Set Utility panel are explained in Chapter 2, “Libraries and Data Sets.” For options A and M, you can specify any DASD-resident sequential or partitioned data set. For the other options, you can specify any DASD-resident data set that is not VSAM.

The following sections describe the options shown on the Data Set Utility panel.

A - Allocate New Data Set

Use option A to allocate a new data set. You must do the following:

- Enter one of the following:
 - An ISPF library name in the PROJECT, GROUP, and TYPE fields
 - An “other” partitioned or sequential data set name in the DATA SET NAME field.

See Chapter 2, “Libraries and Data Sets” if you need help entering the ISPF library name or the data set name.

- If you entered an ISPF library name, the value in the VOLUME SERIAL field is ignored.

Data Set Utility (Option 3.2)

However, if you entered an "other" data set name, you can specify the volume on which to allocate the data set in the VOLUME SERIAL field. You should not enter a volume serial if you want to do one of the following:

- Use the authorized default volume
- Enter a generic unit address in the GENERIC UNIT field on the Allocate New Data Set panel.

See "Volume Serials" on page 32 if you need more information about volume serials.

- If your ISPF libraries and data sets are password-protected, enter the password in the DATA SET PASSWORD field. See "Using Data Set Passwords" on page 49 if you need more information about data set passwords.
- Press the ENTER key.

A panel is displayed to allow you to specify data set allocation values (Figure 41).

```

----- ALLOCATE NEW DATA SET -----
COMMAND ===>

DATA SET NAME: NEWPROJ.MASTER.ASM

VOLUME SERIAL      ===> _          (Blank for authorized default volume) *
GENERIC UNIT       ===>           (Generic group name or unit address) *
SPACE UNITS        ===> TRACK      (BLKS, TRKS, or CYLS)
PRIMARY QUANTITY   ===> 2          (in above units)
SECONDARY QUANTITY ===> 1          (in above units)
DIRECTORY BLOCKS   ===> 1          (Zero for sequential data set)
RECORD FORMAT      ===> FB
RECORD LENGTH      ===> 80
BLOCK SIZE         ===> 3120
EXPIRATION DATE    ===> 87/05/24   (YY/MM/DD,
                                     YY.DDD in Julian form,
                                     DDDD for retention period in days
                                     or blank)

( * Only one of these fields may be specified)

```

Figure 41. Allocate New Data Set Panel - Two-digit Year Support

When you press the ENTER key with this panel displayed, the new data set is allocated and cataloged. Entering the END command returns you to the previous panel without allocating the data set.

An optional installation exit, the data set allocation exit, can control all data set creation, deletion, allocation, and deallocation done directly by PDF. This does not include allocations done by ISPF, the TSO ALLOCATE command, or other TSO commands. See *Installation and Customization* for more information about the data set allocation exit.

Data Set Utility (Option 3.2)

Default values are provided for the fields in Figure 41 on page 109, except for expiration date, based on which of the following occurred most recently:

- What you last entered on this panel
- The last display data set information request (option 3.1, 3.2, or 3.4).

You can type over the displayed defaults if you want to change them. The following is a list of the fields on this panel and their definitions:

VOLUME SERIAL

The volume serial of the direct access volume that is to contain the data set. This field overrides the VOLUME SERIAL field on the Data Set Utility panel. You should not enter a volume serial if you want to do one of the following:

- Use the authorized default volume
- Enter a generic unit address in the GENERIC UNIT field.

If DFP Version 3 Release 1 is installed and available, and the Storage Management Subsystem is active, your data set allocation request can have one of the following possible results if you specify a volume serial:

- If your installation has defined a default storage class, the data set may be allocated on either the specified volume or a different volume.
- If your installation has *not* defined a default storage class, the data set allocation may fail if you specify a volume that is managed by the Storage Management Subsystem.

GENERIC UNIT

The generic unit address for the direct access volume that is to contain the data set, such as 3330, 3350, 3380, and so forth. This field overrides the VOLUME SERIAL field on the Data Set Utility panel. Therefore, you should leave this field blank if you want to do one of the following:

- Use the authorized default volume
- Enter a volume serial in the VOLUME SERIAL field.

Notes:

1. *Leave both the VOLUME SERIAL and GENERIC UNIT fields blank to allow ISPF to select an eligible volume. Eligibility is determined by the "unit" information in your user entry in the TSO User Attribute Data Set (UADS).*
2. *At some installations, you are limited to eligible volumes (see note 1) even when an explicit volume serial is specified. At other installations you can specify any mounted volume. This is an installation option.*

Data Set Utility (Option 3.2)

3. To allocate a data set to a 3850 virtual volume, you must also have MOUNT authority, gained by using the TSO ACCOUNT command.

SPACE UNITS

Any of the following:

TRACK	Tells PDF that the amounts entered in the next two fields are expressed in tracks.
CYLINDER	Tells PDF that the amounts entered in the next two fields are expressed in cylinders.
BLOCK	Tells PDF that the amounts entered in the next two fields are expressed in blocks.

PRIMARY QUANTITY

The primary allocation quantity in tracks, cylinders, or blocks, as indicated in the SPACE UNITS field. This number can be zero for sequential data sets, but must be greater than zero for PDSs. Also, if the primary quantity is zero, the secondary quantity must be greater than zero.

SECONDARY QUANTITY

The secondary allocation quantity in tracks, cylinders, or blocks, as indicated in the SPACE UNITS field. This quantity is allocated when the primary quantity is insufficient.

DIRECTORY BLOCKS

Enter one of the following:

- For *partitioned* data sets, you must specify the number of directory blocks. Each 256-byte block accommodates the following number of directory entries:
 - Data sets with ISPF statistics: 6
 - Data sets without ISPF statistics: 21
 - Load module data sets: 4 to 7, depending on attributes.
- For *sequential* data sets, set this field to zero.

RECORD FORMAT

Any valid combination of the following codes:

F	Fixed-length records.
V	Variable-length records.
U	Undefined format records.
B	Blocked records.
A	ASA printer control characters.

Data Set Utility (Option 3.2)

- | | |
|----------|--|
| M | Machine code printer control characters. |
| S | Standard (for F) or spanned (for V); use only with sequential data sets. |
| T | Track-overflow feature. |

Notes:

1. *You must enter either F, V, or U.*
2. *You can specify S and T, but ISPF does not otherwise support them.*

RECORD LENGTH

The logical record length, in bytes, of the records to be stored in the data set.

BLOCK SIZE

The block size, also called *physical* record length, of the blocks to be stored in the data set. Use this field to specify how many bytes of data to put into each block, based on the record length. For example, if the record length is 80 and the block size is 3120, 39 records can be placed in each block.

Note: The record length and block size are verified to be consistent with the record format, as defined in OS/VS2 MVS Data Management Services Guide. If you need to use non-standard characteristics, use the TSO ALLOCATE command.

EXPIRATION DATE

Lets you protect valuable data by specifying a date, in your national language, when the data set may be deleted. If you try to delete an unexpired data set, PDF displays two panels: a Confirm Delete panel, followed by a Confirm Purge panel. See "D - Delete Entire Data Set" on page 114.1 for more information about deleting unexpired data sets.

Four-digit year support is provided in the MVS/SP Version 2 Release 2 and MVS/XA Data Facility Product (DFP) Version 2 Release 3 operating systems. If you have this support, you will see the panel shown in Figure 42 on page 113 instead of the panel shown in Figure 41 on page 109.

Data Set Utility (Option 3.2)

```

----- ALLOCATE NEW DATA SET -----
COMMAND ===>

DATA SET NAME: NEWPROJ.MASTER.ASM

VOLUME SERIAL      ===> _           (Blank for authorized default volume) *
GENERIC UNIT       ===>             (Generic group name or unit address) *
SPACE UNITS        ===> TRACK      (BLKS, TRKS, or CYLS)
PRIMARY QUANTITY   ===> 2          (In above units)
SECONDARY QUANTITY ===> 1          (In above units)
DIRECTORY BLOCKS   ===> 1          (Zero for sequential data set)
RECORD FORMAT      ===> FB
RECORD LENGTH      ===> 80
BLOCK SIZE         ===> 3120
EXPIRATION DATE    ===> 1987/05/24 (YY/MM/DD, YYYY/MM/DD,
                                   YY.DDD, YYYY.DDD in Julian form,
                                   DDDD for retention period in days
                                   or blank)

( * Only one of these fields may be specified)

```

Figure 42. Allocate New Data Set Panel - Four-digit Year Support

An expiration date is not required, but if you enter one it should be in one of the following formats:

YY/MM/DD, YYYY/MM/DD

Date shown in year, month, and day, or your equivalent national format. The maximum expiration date allowed is:

- 99/12/31 for a two-digit year
- 2155/12/31 for a four-digit year.

YY.DDD, YYYY.DDD

Date shown in Julian format, such as 86.045 for February 14, 1986. The maximum expiration date allowed is:

- 99.365 for a two-digit year
- 2155.365 for a four-digit year.

You can specify a DDD value of up to 366 if the YY or YYYY value represents a leap year.

DDDD The number of days, starting with the creation date, after which the data set can be deleted. DDDD has a range of 0 to 9999.

Data Set Utility (Option 3.2)

C - Catalog Data Set

If you select option C, the specified data set is cataloged. For this option, you must specify the volume serial on which the data set resides, regardless of whether the data set is specified as project, library, and type, or as another data set name. The data set must reside on the specified volume.

* Cataloging Managed Data Sets

The preceding instructions for cataloging data sets do not apply to data sets that reside on Storage Management Subsystem volumes. These data sets are automatically cataloged when you allocate them. They cannot be cataloged by using option C.

R - Rename Entire Data Set

If you select option R, a panel is displayed to allow you to enter the new data set name (Figure 43). Type the new data set name and press the ENTER key to rename, or enter the END command to cancel. Either action returns you to the previous panel.

```

----- RENAME DATA SET -----
COMMAND ==>

DATA SET NAME: ISPFDEMO.XXX.COBOL
VOLUME:       010901

ENTER NEW NAME BELOW:      (The data set will be recataloged.)

ISPF LIBRARY:
  PROJECT ==>
  GROUP   ==>
  TYPE    ==>

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
  DATA SET NAME ==> 'NEWPROJ.DEVEL.COBOL' _

```

Figure 43. Rename Data Set Panel

If you specify a volume serial for a data set to be renamed, PDF checks to see whether the data set is cataloged on that volume. If it is, the rename panel prompts you to specify whether to recatalog the data set. The displayed default for that field is YES. If you specify a volume serial and the data set is not cataloged, it remains uncataloged after you rename it. If a volume serial is not specified, the data set is recataloged to the new data set name and the old data set name is uncataloged.

Note: PDF does not rename VSAM data sets or password-protected data sets.

Data Set Utility (Option 3.2)

Renaming GDG Data Sets

If you are renaming a Generation Data Group (GDG) data set that has been defined with scratch and limit parameters, where the limit has been reached, you must specify a volume serial. The new data set name can then be cataloged with option C.

Also, you cannot catalog more than one version of the same generation of a Generation Data Group (GDG) data set. To rename a GDG data set that is cataloged, first give it a different generation number. Otherwise, the GDG data set is deleted.

* Renaming Managed Data Sets

* When you rename a data set that resides on a Storage Management
* Subsystem volume, you cannot specify a volume serial in the VOLUME
* SERIAL field. Both the catalog entry and the VTOC entry are renamed.

U - Uncatalog Data Set

If you select option U, the specified data set name is uncataloged. There is no need for the specified data set to be allocated or for the volume on which it resides to be mounted.

* Uncataloging Managed Data Sets

* You cannot use option U to uncatalog a data set that resides on a Storage
* Management Subsystem volume. However, the system uncatalogs these
* data sets when you delete them, which is done by using option D of either
* the Data Set utility (3.2) or the Data Set List utility (3.4).

D - Delete Entire Data Set

If you select option D, a confirmation panel (Figure 44 on page 114.2) is displayed so you can make sure you did not select this option by mistake.

If you specify a volume serial for the data set to be deleted, PDF checks to see whether the data set is cataloged on that volume. If so, the Confirm Delete panel prompts you to specify whether to uncatalog the data set. The displayed default is YES. If no volume serial is specified and the data set does not have an expiration date, the data set is deleted and uncataloged.

Note: PDF does not delete VSAM or password-protected data sets.

As directed on the panel, do one of the following:

- Press the ENTER key to confirm the data set deletion.
- Enter the END command to cancel. This action returns you to the previous panel.

If the data set has an expiration date that has not expired, PDF displays a Confirm Purge panel (Figure 45 on page 114.2) after the Confirm Delete panel.

Data Set Utility (Option 3.2)

```

----- CONFIRM DELETE -----
COMMAND ----> _

DATA SET NAME: ISPFDEMO.XXX.COBOL
VOLUME:       010901
CREATION DATE: 1986/11/19

INSTRUCTIONS:

  Press ENTER key to confirm delete request.
  (The data set will be deleted and uncataloged.)

  Enter END command to cancel delete request.

```

Figure 44. Confirm Delete Panel

```

----- CONFIRM PURGE -----
COMMAND ---->

  The data set being deleted has an expiration date which has not expired.

DATA SET NAME: ISPFDEMO.XXX.COBOL
VOLUME:       010901
CREATION DATE: 1986/11/19
EXPIRATION DATE: 1999/12/31

PURGE DATA SET ----> _      (YES or NO)

INSTRUCTIONS:

  Enter YES to confirm the purge request.
  (A request will be issued for the data set
  to be deleted and uncataloged.)

  Enter NO or END command to cancel the purge request.

```

Figure 45. Confirm Purge Panel

This panel contains the following field:

PURGE DATA SET

Tells PDF whether to purge the data set. The valid values are:

YES Tells PDF to purge the data set.

NO Tells PDF *not* to purge the data set.

The statement that is enclosed in parentheses on the Confirm Purge panel shows whether the data set to be purged will be cataloged or uncataloged.

S - Data Set Information (Short)

If you select option S, information about the selected data set is displayed. The information displayed by option S is the same information displayed by option S on the Library Utility panel (option 3.1). See Figure 39 on page 106 for an example. To return to the previous panel, press the ENTER key or enter the END command.

The 1st extent quantity, secondary quantity, current allocation, and current utilization sizes are shown in tracks for data sets that are allocated in bytes, kilobytes, or megabytes on a volume that is *not* managed by the Storage Management Subsystem.

If the data set was allocated on a Storage Management Subsystem volume, the data set information panel shows the management class, storage class, and data class for that data set. Also, the 1st extent quantity, secondary quantity, current allocation, and current utilization sizes can be shown in bytes, kilobytes, or megabytes, in addition to tracks, blocks, or cylinders. See “Information (Short) for Managed Data Sets” on page 107 if you need more information.

Blank - Data Set Information

If you leave the OPTION field blank, information about the selected data set is displayed. The information displayed is the same information displayed by option I on the Library Utility panel (option 3.1). See Figure 38 on page 104 for an example. To return to the previous panel, press the ENTER key or enter the END command.

The 1st extent quantity, secondary quantity, current allocation, and current utilization sizes are shown in tracks for data sets that are allocated in bytes, kilobytes, or megabytes on a volume that is *not* managed by the Storage Management Subsystem.

If the data set was allocated on a Storage Management Subsystem volume, the data set information panel shows the management class, storage class, and data class for that data set. Also, the 1st extent quantity, secondary quantity, current allocation, and current utilization sizes can be shown in bytes, kilobytes, or megabytes, in addition to tracks, blocks, or cylinders. See “Information for Managed Data Sets” on page 105 if you need more information.

*** M - Enhanced Data Set Allocation**

Use option M to allocate a new data set by using the Storage Management Subsystem classes (management class, storage class, and data class). A data set that is allocated on a volume that is managed by the Storage Management Subsystem is referred to in this book as a *managed* data set. Conversely, a data set that is allocated on a volume that is *not* managed by the Storage Management Subsystem is referred to as a *non-managed* data set.

Data Set Utility (Option 3.2)

To use option M, you must do the following:

- Enter one of the following:

- An ISPF library name in the PROJECT, GROUP, and TYPE fields
- An “other” partitioned or sequential data set name in the DATA SET NAME field.

See Chapter 2, “Libraries and Data Sets” if you need help entering the ISPF library or data set name.

- If you fill in the VOLUME SERIAL field, PDF copies the volume serial that you specify to the Allocate New Data Set panel. See VOLUME SERIAL on page 114.6 if you need more information about using volume serials.

Note: You cannot assign a password to a managed data set. Therefore, the DATA SET PASSWORD field is ignored when you use option M. See “Using Data Set Passwords” on page 49 if you need more information about data set passwords.

- Press the ENTER key.

A panel is displayed to allow you to specify data set allocation values (Figure 45.1).

```

----- ALLOCATE NEW DATA SET -----
COMMAND ---->
DATA SET NAME: FBROWN.PRIVATE.PANELS

MANAGEMENT CLASS  ----> M1_      (Blank for default management class)
STORAGE CLASS     ----> S1       (Blank for default storage class)
VOLUME SERIAL     ----> 101987   (Blank for authorized default volume)
DATA CLASS        ----> D1       (Blank for default data class)
SPACE UNITS       ----> KB       (BLKS, TRKS, CYLS, KB or MB)
PRIMARY QUANTITY  ---->          (In above units)
SECONDARY QUANTITY ---->          (In above units)
DIRECTORY BLOCKS  ---->          (Zero for sequential data set)
RECORD FORMAT     ---->
RECORD LENGTH     ---->
BLOCK SIZE        ---->
EXPIRATION DATE   ----> 88/03/15 (YY/MM/DD,
                                YY.DDD in Julian form,
                                DDDD for retention period in days
                                or blank)

```

Figure 45.1. Allocate New Data Set Panel - Managed Data Set Support

When you press the ENTER key with this panel displayed, the new data set is allocated and cataloged. Entering the END command returns you to the previous panel without allocating the data set.

An optional installation exit, the data set allocation exit, can control all data set creation, deletion, allocation, and deallocation done directly by PDF. This exit, however, does not control allocations done by ISPF, the

Data Set Utility (Option 3.2)

TSO ALLOCATE command, or other TSO commands. See *Installation and Customization* for more information about the data set allocation exit.

Your installation must use the Data Facility Product (DFP) Version 3 Release 1 to define the values that you enter in the MANAGEMENT CLASS, STORAGE CLASS, and DATA CLASS fields. If you have no specific requirements, you can leave these fields blank. However, you should be aware that your installation may provide default management, storage, and data classes. These defaults would take effect if you leave any of the class fields blank and may even override any classes that you specify.

The values in the MANAGEMENT CLASS, STORAGE CLASS, and DATA CLASS fields are the only values from this panel that are saved in your user profile. Therefore, they are the only fields PDF remembers from session to session. You can type over the displayed values if you want to change them.

MANAGEMENT CLASS

Used to obtain data management-related information (migration, backup, and retention criteria, such as expiration date) for the data set allocation.

If you have no specific management class requirements, you can leave this field blank. However, you should be aware that your installation may provide a default management class. This default may even override any management class that you specify.

STORAGE CLASS

Used to obtain the storage-related information (VOLUME SERIAL or GENERIC UNIT) for the data set allocation. Any volume serial that you enter in the VOLUME SERIAL field is ignored unless the storage class that you use includes the GUARANTEED SPACE = YES attribute. See VOLUME SERIAL if you need more information.

DATA CLASS

Used to obtain the data-related information (SPACE UNITS, PRIMARY QUANTITY, SECONDARY QUANTITY, DIRECTORY BLOCKS, RECORD FORMAT, and RECORD LENGTH) for the allocation of the data set.

The remaining fields in the panel shown in Figure 45.1 on page 114.4 are the same as those on the Allocate New Data Set panel that appears when you select option A (Figure 41 on page 109). Exceptions that pertain to the usage and display of those fields are listed below.

If you enter any values in those fields, those values may replace values specified by a class definition. Be careful when you do this, because you may allocate a data set that has inconsistent attributes. See "Allocation Errors" on page 114.7 if you need more information.

Data Set Utility (Option 3.2)

VOLUME SERIAL

This field is one that you probably will not need to use very often. It is not required and is usually ignored by the Storage Management Subsystem. If you leave this field blank, the data set is allocated on one of the following:

- The authorized default volume, if the data set you are allocating is not managed by the Storage Management Subsystem. The data set is not managed if you do not specify a storage class or if a default storage class is not used.
- The volume specified by the storage class you are using, whether that storage class is specified by you or is a default storage class selected by the system.

When a storage class is used, your installation and the Storage Management Subsystem assume joint responsibility for determining the volume on which the data set is allocated. The Storage Management Subsystem lets the installation select the volumes that are eligible to contain the data set; it then chooses one of those volumes and allocates the data set. The Storage Management Subsystem's volume choice is based on 1) storage requirements that are defined in the storage class and 2) the amount of space a volume has available.

If you specify a volume serial, the data set *may* be allocated on that volume. However, the volume that you specify must be managed by the Storage Management Subsystem, which means:

- You must use a storage class when allocating the data set. If you leave the STORAGE CLASS field blank, a default storage class must be available.
- The storage class that you use must include the GUARANTEED SPACE= YES attribute. When GUARANTEED SPACE= YES and you specify a volume serial, the Storage Management Subsystem attempts to honor your request. The request fails if the installation does not select the volume or if the volume does not have enough available space.

See "Volume Serials" on page 32 for general information about volume serials.

GENERIC UNIT

The GENERIC UNIT field is not used by option M. Therefore, it is not displayed.

SPACE UNITS

In addition to the TRACK, CYLINDER, and BLOCK space units (see SPACE UNITS on page 111), this panel lets you specify:

- KB** Tells PDF that the primary and secondary quantities are expressed in kilobytes.

Data Set Utility (Option 3.2)

MB Tells PDF that the primary and secondary quantities are expressed in megabytes.

* Allocation Errors

PDF attempts to recognize inconsistent attributes for partitioned and sequential data sets before allocating them. However, the Allocation Error panel (Figure 45.2) is displayed when conditions that PDF has no control over allow the allocation of such a data set. These conditions are caused by:

- A data class that specifies inconsistent attributes
- Attributes entered on the Allocate New Data Set panel that create inconsistency by overriding other attributes specified by the data class.

----- ALLOCATION ERROR -----

COMMAND ===> _

DATA SET NAME: SYSTEM.LOAD

SPECIFIED DATA SET HAS ZERO BLOCK SIZE

The data set allocated contains inconsistent attributes as indicated by the message displayed above. Prior to allocating a managed data set, PDF cannot always determine if the attributes are inconsistent. The data class used when allocating the data set may contain inconsistent attributes, or the attributes you specified on the allocation panel may conflict with those defined in the data class. This panel gives you the opportunity to delete this data set. If you keep the data set, other PDF functions, such as edit, move, copy, or promote, may not be able to use the data set.

INSTRUCTIONS:

Press the ENTER key to DELETE the data set.

Enter the END command to KEEP the data set.

Figure 45.2. Allocation Error Panel

The term *inconsistent attributes* refers to incompatible values that have been specified for one or more of the following:

- Space units
- Primary quantity
- Secondary quantity
- Directory blocks
- Record format
- Record length
- Block size.

For example, if you allocate a data set with an undefined record format (RECFM=U) and a block size of zero (BLOCKSIZE=0), certain PDF functions and services cannot use the data set. These functions and services include:

- Move, copy, promote, and LMF functions
- LMMOVE, LMCOPY, LMPROM, and LMINIT services.

Data Set Utility (Option 3.2)

* However, when either the linkage editor or the IEBCOPY utility has been
* invoked, these functions and services determine the best block size for the
* data set. Then, when the data set has a block size greater than zero, the
* PDF functions and services listed above can be used.

* The Allocation Error panel gives you the opportunity to delete such a data
* set because other PDF functions, such as Browse (option 1) and Edit
* (option 2), may not be able to use it.

* The sample panel (Figure 45.2 on page 114.7) shows this error message
* example:

* SPECIFIED DATA SET HAS ZERO BLOCK SIZE

* Press the ENTER key to delete the data set or the END key to keep the
* data set.

Data Set List Utility (Option 3.4)

Track values do not include the remaining alternate tracks for the volume. The free space track values are the number of tracks for the free cylinders plus any additional free tracks.

The fields shown on the VTOC display are:

UNIT Shows the type of DASD device the volume is on, such as 3330, 3350, 3380, and so forth.

VOLUME DATA

Describes general information about the volume:

TRACKS	Total tracks on the volume.
%USED	Percentage of total tracks not available for allocation.
TRKS/CYLS	Number of tracks per cylinder for this volume.

VTOC DATA

Describes general information about the VTOC on the volume:

TRACKS	Total tracks allocated to the VTOC.
%USED	Percentage of allocated tracks used by data set control blocks (DSCBs).
FREE DSCBS	Number of unused DSCBs.

FREE SPACE

Describes the free space available for data set allocation on the volume under the headings **TRACKS** and **CYLS**, showing:

SIZE Total number of free tracks and cylinders.

LARGEST

The largest number of contiguous free tracks and cylinders.

FREE EXTENTS

The number of free areas with free cylinders.

PV - Print VTOC Information

Use option PV to print VTOC information. The List VTOC utility (formerly option 3.7) has been removed from PDF. Options V and PV do the same function formerly provided by this utility.

To use option PV, you must do the following:

- Blank out the DSNAME LEVEL field.
- In the VOLUME field, specify the volume serial for which you want PDF to print information.

Data Set List Utility (Option 3.4)

Note: VTOC information is formatted the same, whether displayed or printed. Therefore, values entered in the INITIAL DISPLAY VIEW field have no effect when using option PV.

- Press the ENTER key to print the VTOC information. The VTOC information is stored in the ISPF list data set.

Data Set List Utility Line Commands

After you display a data set list by leaving the OPTION field blank, you can enter a line command to the left of the data set name. You can also enter TSO commands or CLIST names. The “/” symbol, which can be used with TSO commands and CLISTs, can also be used with the B (Browse data set), E (Edit data set), and M (Display member list) line commands to specify a member name or a pattern, such as /(MEM1) or /(MEM*), respectively. You can type over the Data Set Name field to enter commands that require more than the space provided. For more information about using this symbol, see “Using the Slash (/) Symbol” on page 132.

The available line commands are shown at the bottom of the Data Set List Utility panel and described below:

B - Browse Data Set

Processing is the same as with the Browse option (1), except that mixed mode is the assumed operation mode and you cannot specify a data set format. To browse a data set with mixed mode off, you must use the Browse option and enter NO in the MIXED MODE field on the Browse - Entry Panel. If you enter the B line command beside an ISPF library or other partitioned data set, a Browse member list is displayed. See “Using Member Selection Lists” on page 37 for more information.

E - Edit Data Set

For the E command, the processing is similar to selecting the Edit option (2) and entering the library or data set name on the Edit - Entry Panel, except that:

- Mixed mode is the assumed operation mode
- You cannot edit while in recovery pending status
- You cannot specify a data set format, an edit profile, or an initial macro.

If you select a library or other partitioned data set, an Edit member list is displayed. See “Using Member Selection Lists” on page 37 for more information.

Data Set List Utility (Option 3.4)

D - Delete Data Set

For the D command, the processing is the same as if you had selected option D from the Data Set utility (3.2). This command deletes the complete data set.

If you specify YES in the CONFIRM DELETE REQUEST field on the Data Set List Utility panel, the Confirm Delete panel (Figure 44 on page 114.2) is displayed to allow you to continue or cancel the operation. If you are trying to delete an unexpired data set, the Confirm Purge panel (Figure 45 on page 114.2) is also displayed.

R - Rename Data Set

For the R command, the processing is the same as if you had selected option R from the Data Set utility (3.2). The Rename Data Set panel (Figure 43 on page 114) is displayed to let you specify the new name.

Note: PDF does not rename VSAM data sets or password-protected data sets.

When you rename a data set that resides on a Storage Management Subsystem volume, you cannot specify a volume serial in the VOLUME SERIAL field. Both the catalog entry and the VTOC entry are renamed.

I - Data Set Information

For the I command, the processing is the same as if you had selected option I from the Library utility (3.1) or left the OPTION field blank with the Data Set utility (3.2).

The 1st extent quantity, secondary quantity, current allocation, and current utilization sizes are shown in tracks for data sets that are allocated in bytes, kilobytes, or megabytes on a volume that is *not* managed by the Storage Management Subsystem.

If the data set was allocated on a Storage Management Subsystem volume, the data set information panel shows the management class, storage class, and data class for that data set. Also, the 1st extent quantity, secondary quantity, current allocation, and current utilization sizes can be shown in bytes, kilobytes, or megabytes, in addition to tracks, blocks, or cylinders. See "Information for Managed Data Sets" on page 105 if you need more information.

S - Information (Short)

For the S command, the processing is the same as if you had selected option S from the Library utility (3.1) or the Data Set utility (3.2).

The 1st extent quantity, secondary quantity, current allocation, and current utilization sizes are shown in tracks for data sets that are allocated in bytes, kilobytes, or megabytes on a volume that is *not* managed by the Storage Management Subsystem.

Data Set List Utility (Option 3.4)

If the data set was allocated on a Storage Management Subsystem volume, the data set information panel shows the management class, storage class, and data class for that data set. Also, the 1st extent quantity, secondary quantity, current allocation, and current utilization sizes can be shown in bytes, kilobytes, or megabytes, in addition to tracks, blocks, or cylinders. See "Information (Short) for Managed Data Sets" on page 107 if you need more information.

C - Catalog Data Set

For the C command, the processing is the same as if you had selected option C from the Data Set utility (3.2).

You cannot use the C command to catalog a data set on a Storage Management Subsystem volume. These data sets are automatically cataloged when you allocate them.

U - Uncatalog Data Set

For the U command, the processing is the same as if you had selected option U from the Data Set utility (3.2).

You cannot use option U to uncatalog a data set that resides on a Storage Management Subsystem volume. However, the system uncatalogs these data sets when you delete them, which is done by using option D of either the Data Set utility (3.2) or the Data Set List utility (3.4).

P - Print Data Set

For the P command, the processing is the same as if you had selected option L from the Library utility (3.1). This command formats the contents of a source data set for printing and records the output in the ISPF list data set. It also produces an index listing, which appears at the beginning of the output. Figure 185 on page 340 shows a sample source listing.

X - Print Index Listing

For the X command, the processing is the same as if you had selected option X from the Library utility (3.1). The index listing is recorded in the ISPF list data set.

An index listing shows general data set information and member names for partitioned data sets. For ISPF source libraries, activity statistics are listed for each member. For load libraries, load module information is listed for each member. In both of these types of listings, the 1st extent quantity, secondary quantity, current allocation, and current utilization sizes are shown in tracks for data sets that are allocated in bytes, kilobytes, or megabytes on a volume that is *not* managed by the Storage Management Subsystem. See Figure 186 on page 341 and Figure 187 on page 342.1, respectively, for examples.

For data sets on Storage Management Subsystem volumes, the management, storage, and data classes used are shown under the GENERAL DATA heading. Also, the 1st extent quantity, secondary quantity, current allocation, and current utilization sizes can be shown in bytes, kilobytes, or

Data Set List Utility (Option 3.4)

* megabytes, in addition to tracks, blocks, or cylinders. See Figure 186.1 on
 * page 342 and Figure 187.1 on page 342.2, respectively, for examples.

M - Display Member List

The M command displays a member selection list of a partitioned data set. This member list provides an expanded line command field in the area to the left of the list. The line command field in other member lists has room for only one character.

From the member list, you can use the same primary commands and line commands that are valid for Library utility (3.1) member selection lists, plus the E (Edit data set member) line command. See "Using Member Selection Lists" on page 37 and "Library and Data Set List Utility Line Commands" on page 47 if you need more information.

You can also enter TSO commands or CLISTs in the line command field. If you enter a line command other than B (browse), D (delete), E (edit), R (rename), or P (print), PDF interprets it as a TSO command or CLIST.

The command is selected with the quoted, fully-qualified data set name and member added to it, such as:

```
COMMAND 'DSN(MEMBER)'
```

See "TSO Commands and CLISTs" on page 132 if you need more information.

Note: If the TSO command or CLIST issues a return code greater than or equal to 8, processing stops and an error message is displayed.

Figure 60 shows an example of a member list with statistics and an expanded line command field. This panel contains the same information as Figure 12 on page 40, except the "CREATED" field is deleted.

DSLIST -- FBROWN.PRIVATE.PANELS -----										ROW 00001 OF 00020	
COMMAND =====>										SCROLL =====> PAGE	
NAME	RENAME	LIB	VV	MM	CHANGED	SIZE	INIT	MOD	ID		
ISRB00		3	21.11	83/12/05	09:12	29	23	0	DONLEY		
ISRB01		2	20.04	85/12/13	08:51	29	23	0	MARTIN		
ISRB10		3	21.00	84/03/26	10:53	29	29	0	PATTEN		
ISRB15		3	20.03	83/12/05	09:12	26	23	0	BROWN		
ISRB16		3	20.01	82/11/05	09:24	20	20	0	KISH		
ISRB18		2	21.01	86/01/21	14:03	17	11	0	BERNACK		
ISRB50		3	21.00	84/03/26	10:53	24	24	0	WALTERS		
ISRC00		1	ALIAS								
ISRC60		1	21.02	83/11/10	11:48	11	8	0	ROGERS		
ISRE00		3	21.01	83/02/04	11:05	26	26	0	SPENCER		
ISRE01		3	21.04	83/07/18	09:58	29	29	0	RICHEY		
ISRE02		3	21.11	83/07/18	10:00	15	8	0	GURLEY		
ISRE03		3	21.04	83/07/18	10:02	29	26	0	DUFF		
ISRE04		3	20.16	83/06/22	11:03	29	44	0	COONS		
ISRE05		2	01.01	85/10/04	11:13	29	29	0	RUPKAL		
ISRE06		3	20.10	84/02/17	14:31	29	29	0	MANNINO		
ISRE07		3	20.02	85/06/17	14:56	29	29	0	KELLMA		
ISRE08		2	20.01	86/01/20	09:04	23	20	0	MCINDOO		
ISRE09		2	21.13	85/12/13	08:52	29	1	0	DIRKSE		
ISRE10		3	20.03	83/12/05	09:13	29	29	0	SEABOLD		
END											

Figure 60. Member List Display - Expanded Line Command Field

Data Set List Utility (Option 3.4)

Figure 61 on page 141 shows load module library statistics with an expanded line command field. This panel contains the same information shown in Figure 13 on page 41, except the "ALIAS-OF" field is deleted.

Source and Index Listings

Index Listings

PDF provides index listings at your request through the X (Print index listing) option of the Library utility (3.1) or the Data Set List utility (3.4).

Index Listings for Source Libraries

Figure 186 shows a sample index listing for an ISPF library.

PROJECT: ISPFDEMO							DATE: 86/02/07
GROUP: MYLIB							TIME: 17:22
TYPE: PLI							PAGE: 001
GENERAL DATA:				CURRENT ALLOCATION:		CURRENT UTILIZATION:	
VOLUME SERIAL:	D00163	RECORD FORMAT:	FB	20 TRACKS		5 TRACKS	
DEVICE TYPE:	3330	RECORD LENGTH:	80	1 EXTENT		1 EXTENT	
ORGANIZATION:	PO	BLOCK SIZE:	1,600	10 DIRECTORY BLOCKS		2 DIRECTORY BLOCKS	
CREATION DATE:	85/12/08	1ST EXTENT SIZE:	20			10 MEMBERS	
EXPIRATION DATE:	**NONE**	SECONDARY QUAN:	0				
MEMBER NAME	VERS. MOD LEVEL	CREATION DATE	DATE AND TIME LAST MODIFIED	CURRENT NO. LINES	INITIAL NO. LINES	MODIFIED NO. LINES	USER ID
ACCOUNT	01.00	86/01/10	86/01/10 17:07	21	21	0	HUNTER
ACCT1	01.01	86/01/09	86/01/23 14:52	99	193	0	BECKETT
ACCT2	01.00	86/01/09	86/01/10 17:07	20	20	0	BECKETT
COINS	01.04	86/01/24	86/01/25 16:20	19	19	4	JOSLIN
COMPX	01.00	86/01/09	86/01/10 17:08	44	44	0	JOSLIN
COMPY	01.01	86/01/14	86/01/15 12:30	13	13	1	FISHER
DCLS	01.00	86/01/23	86/01/24 15:14	20	20	0	SAMLEE
LISTOUT	01.02	86/01/23	86/01/24 15:00	17	13	6	SAMLEE
MAIN	01.00	86/01/09	86/01/10 17:08	4	4	0	KIRK
TESTDIR	01.02	86/01/23	86/02/06 17:04	30	43	10	LAIDLAW
MAXIMUMS:	01.04	86/04/24	86/05/06 17:04	99	193	10	
TOTALS:				287	390	21	
END OF MEMBER LIST							

Figure 186. Sample Index Listing - Source Library

The heading information includes:

- Project, library, and type
- Date and time the listing was produced
- Page number.

This is followed by general information about the data set, including current space allocation and utilization. The 1st extent quantity, secondary quantity, current allocation, and current utilization sizes are shown in tracks for data sets that are allocated in bytes, kilobytes, or megabytes on a volume that is *not* managed by the Storage Management Subsystem. Following this, the member name and statistics are printed for each member in the data set, arranged in alphabetic order. For sequential data sets, the index listing contains only the general information.

Source and Index Listings

* Index Listings for Managed Source Libraries

* Figure 186.1 shows a sample index listing for an ISPF library.

PROJECT: ISPFDEMO

GROUP: MYLIB

TYPE: PANELS

DATE: 87/12/03

TIME: 15:44

PAGE: 001

GENERAL DATA:

MANAGEMENT CLASS: M1

STORAGE CLASS: S1

DATA CLASS: D1

VOLUME SERIAL: D00163

DEVICE TYPE: 3380

ORGANIZATION: PO

CREATION DATE: 86/08/22

EXPIRATION DATE: **NONE**

GENERAL DATA:

RECORD FORMAT: FB

RECORD LENGTH: 80

BLOCK SIZE: 6,160

1ST EXTENT SIZE: 7,162,640

SECONDARY QUAN: 7,168,000

CURRENT ALLOCATION:

7,172,640 KILOBYTES

1 EXTENT

35 DIRECTORY BLOCKS

CURRENT UTILIZATION:

2,235 KILOBYTES

2 EXTENT

10 DIRECTORY BLOCKS

8 MEMBERS

MEMBER NAME	TTR (HEX)	VERS. MOD LEVEL	CREATION DATE	DATE AND TIME LAST MODIFIED	CURRENT NO. LINES	INITIAL NO. LINES	MODIFIED NO. LINES	USER ID
ISRUADCS	000025	21.46	81/06/10	87/11/19 09:42	37	23	0	Z73BDR
ISRUAPS	000103	21.23	84/12/07	87/12/01 11:06	48	33	0	Z73BDR
ISRUASO	000105	21.56	81/06/10	87/12/01 11:06	48	48	0	Z73BDR
ISRUASX	000107	21.44	81/06/10	87/12/01 11:06	50	50	0	Z73BDR
ISRU1734	000027	21.06	81/06/26	87/11/30 13:46	27	25	0	Z73BDR
ISRU1735	000101	21.06	81/06/26	87/11/30 13:49	26	25	0	Z73BDR
ISRU1739	000029	21.09	81/06/26	87/11/30 13:49	27	25	0	Z73BDR
MIKIE	000109	01.92	87/01/24	87/12/01 12:48	42	41	0	Z73BDR
MAXIMUMS:		21.56	87/01/24	87/12/01 12:48	50	50	0	
TOTALS:					305	270	0	

END OF MEMBER LIST

* Figure 186.1. Sample Index Listing - Managed Source Library

* The only differences between this index listing and the one shown in Figure 186 on page 341 are the following:

- * Management, storage, and data classes are shown under the GENERAL DATA heading.
- * The 1st extent size, secondary quantity, current allocation, and current utilization sizes can be shown in bytes, kilobytes, or megabytes, in addition to tracks, blocks, or cylinders.

Index Listings for Load Libraries

An index listing for an OBJ library is similar to an index listing for a source library, except that no statistics are maintained. A sample index listing for a LOAD library is shown in Figure 187. Here, the module attributes are printed to the right of each member name.

```

PROJECT: ISPFDEMO
GROUP: MYLIB
TYPE: LOAD
DATE: 86/02/07
TIME: 17:22
PAGE: 001

GENERAL DATA:
VOLUME SERIAL: 000172
DEVICE TYPE: 3330
ORGANIZATION: PO
CREATION DATE: 85/12/08
EXPIRATION DATE: **NONE**

GENERAL DATA:
RECORD FORMAT: U
RECORD LENGTH: 0
BLOCK SIZE: 3,072
1ST EXTENT SIZE: 44
SECONDARY QUAN: 4

CURRENT ALLOCATION:
88 BLOCKS
12 EXTENTS
4 DIRECTORY BLOCKS

CURRENT UTILIZATION:
86 BLOCKS
12 EXTENTS
2 DIRECTORY BLOCKS
8 MEMBERS

MEMBER NAME ALIAS OF SSI LOAD MODULE SIZE (HEX) (DEC) ENTRY POINT (HEX) TTR (HEX) AUTH CODE MODULE ----- ATTRIBUTES -----
ISPFEDIT 000148 328 000000 000006 00 FO RF RN RU
ISPFMAIN 004F98 20,376 000000 000000 00 FO RF RN RU
ISPFMO 0016E8 5,864 000000 000204 00 FO RF RN RU
ISPFSUBS 024698 149,144 000000 000211 00 FO RF RN RU
ISPFIBLS 002470 9,328 000000 00110B 00 FO
ISPFUDA 0046D8 18,136 000000 00120A 00 FO RF RN RU
ISPFURS 000FC0 4,032 000000 0014CA 00 FO RF RN RU
ISPF3277 000548 1,352 000000 001504 00 FO RF RN RU

TOTALS: 00032EB0 208,560

END OF MEMBER LIST

-----
MODULE ATTRIBUTE CODES:
FO PROCESS ONLY BY F LEVEL LINKAGE EDITOR
NX NOT EXECUTABLE
OL ONLY LOADABLE
OV IN OVERLAY STRUCTURE
RF REFRESHABLE
RN REENTERABLE
RU REUSABLE
SC SCATTER FORMAT
TS MODULE TO BE TESTED

```

Figure 187. Sample Index Listing - Load Library

As in the source library index listing, the 1st extent quantity, secondary quantity, current allocation, and current utilization sizes are shown in tracks for data sets that are allocated in bytes, kilobytes, or megabytes on a volume that is *not* managed by the Storage Management Subsystem.

Source and Index Listings

* Index Listings for Managed Load Libraries

Figure 187.1 shows a sample index listing for a load library.

```

PROJECT: ISPFDEMO
GROUP: MYLIB
TYPE: LOAD

GENERAL DATA:
MANAGEMENT CLASS: M1
STORAGE CLASS: S1
DATA CLASS: D1
VOLUME SERIAL: 000163
DEVICE TYPE: J380
ORGANIZATION: PO
CREATION DATE: 86/08/22
EXPIRATION DATE: **NONE**

GENERAL DATA:
RECORD FORMAT: FB
RECORD LENGTH: 80
BLOCK SIZE: 6,160
1ST EXTENT SIZE: 7,162,640
SECONDARY QUAN: 7,168,000

CURRENT ALLOCATION:
7,172,640 KILOBYTES
1 EXTENT
35 DIRECTORY BLOCKS

CURRENT UTILIZATION:
2,235 KILOBYTES
2 EXTENT
10 DIRECTORY BLOCKS
8 MEMBERS

MEMBER      ALIAS  SSI    LOAD MODULE  SIZE  ENTRY POINT  TTR    AUTH  MODULE
NAME        OF      OF      (HEX)      (DEC)  (HEX)      (HEX)  CODE  ----- ATTRIBUTES -----

ISPFEDIT      000148      328      000000      000006      00      FO      RF RN RU
ISPFMAIN      004F98     20,376     000000      000000      00      FO      RF RN RU
ISPFMD        0016E8      5,864     000000      000204      00      FO      RF RN RU
ISPFSUBS      024698     149,144     000000      000211      00      FO      RF RN RU
ISPFIBLS      002470      9,328     000000      00110B      00      FO
ISPFUDA       0046D8     18,136     000000      00120A      00      FO      RF RN RU
ISPFURS       000FC0      4,032     000000      00140A      00      FO      RF RN RU
ISPF3277      000548      1,352     000000      001504      00      FO      RF RN RU

TOTALS:      00032E80     208,560

END OF MEMBER LIST

-----
MODULE ATTRIBUTE CODES:
FO      PROCESS ONLY BY F LEVEL LINKAGE EDITOR
NX      NOT EXECUTABLE
OL      ONLY LOADABLE
OV      IN OVERLAY STRUCTURE
RF      REFRESHABLE
RN      REENTERABLE
RU      REUSABLE
SC      SCATTER FORMAT
TS      MODULE TO BE TESTED

```

* Figure 187.1. Sample Index Listing - Managed Load Library

The only differences between this index listing and the one shown in Figure 187 on page 342.1 are the following:

- Management, storage, and data classes are shown under the GENERAL DATA heading.
- The 1st extent size, secondary quantity, current allocation, and current utilization sizes can be shown in bytes, kilobytes, or megabytes, in addition to tracks, blocks, or cylinders.

Index

Special Characters

.IM control word 211
 < FIELD > statement 164
 (ACCOUNT) parameter
 JOB STATEMENT INFORMATION 56
 | (OR symbol) 15
 & (command prefix)
 displaying commands after execution 87
 &xxxxx action
 ACTION column 161
 * (comment line) process statement
 Extended Search-For utility (option 3.13.S) 401
 SuperCE utility (option 3.13) 401
 * (in source listing) 340
 QUAL field 320
)ATTR section 164
)BODY section 164
)INIT section 164
 / (data set name character)
 Data Set List utility (option 3.4) 132
 // * lines
 JOB STATEMENT INFORMATION 56
 //OUTDD DD field 178
 % field 130
 %USED field 137
 %USED keyword
 SORT command 146
 ? (invalid load module directory fields) 42
 ? (more than 16 extents)
 FREE EXTENTS field 137
 'NAME' parameter
 JOB STATEMENT INFORMATION 56
 = (jump function) 13, 25
 = (repeat last command) line command
 Data Set List utility (option 3.4) 141
 "Target" Panel for Promoting 121
 "To" Panel for Copying 116
 "To" Panel for Moving 118
 [] (optional operands) 15
 { } (one operand required) 15

A

A (ANSI) parameter
 PRINTER CARRIAGE CONTROL field 156
 A (ASA printer characters) record format
 RECORD FORMAT field 114
 A (variable attributes) field 294, 300, 302, 304

AC field 42
 ACCOUNT command
 TSO 32, 111
 ACCOUNT parameter
 JOB STATEMENT INFORMATION 56
 ACTION column 160
 ACTION field 121
 Activate/Create Profile DS Name field 197
 activating library controls 34
 ACTIVE field 316, 318, 320
 Add Row panel 304
 Additional IBM Program Development Products
 panel 329
 ADDITIONAL INPUT LIBRARIES field 216
 AFTER field 324
 AFTER parameter
 Breakpoint Primary Option Menu 324
 WHEN field 320
 ALARM indicator 165
 ALIAS action
 ACTION column 161
 alias as substitute for PDF 4, 5
 ALIAS-OF field 42
 ALL parameter
 FIND command 91
 FUNCTION field 316
 VARIABLE field 317
 ALLOCATE command
 TSO 109, 114.5
 Allocate New Data Set panel
 four-digit year support 113
 managed data set support 114.4
 two-digit year support 109
 allocating libraries 29, 33
 Allocation Error panel 114.7
 American National Standard Institute 156
 ANSI
 See American National Standard Institute
 ANYC process option
 Extended Search-For utility (option 3.13.S) 379
 SuperCE utility (option 3.13) 379
 APNDLST process option
 Extended Search-For utility (option 3.13.S) 381
 SuperCE utility (option 3.13) 381
 APNDUPD process option
 SuperCE utility (option 3.13) 383
 APPLICATION field 324
 APPLICATION ID field 159
 ASIS parameter
 COMMAND LINE PLACEMENT field 74
 Assembler 216
 Assembler H (option 4.1) 219
 allocation data sets

SYSIN 367
 SYSLIB 367
 SYSLIN 368
 SYSPRINT 368
 SYSPUNCH 369
 SYSTEM 368
 SYSUT1 369
 Assembler H (option 5.1) 258
 ASSEMBLER OPTIONS field
 Foreground Assembler H (option 4.1) 219
 Foreground Assembler XF (option 4.1A) 220
 Assembler XF (option 4.1A) 220
 Assembler XF (option 5.1A) 260
 asterisk (*) parameter
 SELECT command 45
 ATTENTION (PA1) key 18
 ATTRIB display view
 INITIAL DISPLAY VIEW field 126
 ATTRIBUTES field 42

B

B (batch) parameter
 MODE field 173
 B (blocked) record format
 RECORD FORMAT field 111
 B (browse data set) line command
 Data Set List utility (option 3.4) 138
 B (browse) line command
 Data Set List utility (option 3.4) 47
 Library utility (option 3.1) 47
 Batch (option 5) 251
 Assembler H (option 5.1) 258
 Assembler XF (option 5.1A) 260
 JCL generation for Assembler and Linkage Editor 257
 JCL generation for compilers 256
 Linkage Editor (option 5.7) 270
 Member Parts List (option 5.12) 273
 OS/VS COBOL compiler (option 5.2A) 264
 output listings 54
 overview 6
 Pascal/VS compiler (option 5.6) 269
 password protection, lack of 253
 PL/I checkout compiler (option 5.4) 267
 PL/I optimizing compiler (option 5.5) 268
 processing sequence 251
 VS COBOL II compiler (option 5.2) 262
 VS COBOL II interactive debug (option 5.10) 272
 VS FORTRAN compiler (option 5.3) 265
 Batch Assembler H panel 258
 Batch Assembler XF panel 260
 Batch Linkage Edit panel 270
 Batch Member Parts List panel 273
 Batch OS/VS COBOL Compile panel 264
 Batch Pascal/VS Compile panel 269
 Batch PL/I Checkout Compile panel 267

Batch PL/I Optimizing Compile panel 268
 Batch Selection Panel 251
 Batch Selection Panel With JCL Generated 255
 Batch VS COBOL II Compile panel 262
 Batch VS FORTRAN Compile panel 265
 BEFORE field 324
 BEFORE parameter
 Breakpoint Primary Option Menu 324
 WHEN field 320
 Blank action
 ACTION column 161
 blank options 9
 blank parameter
 ACTIVE field 316, 318, 320
 DSORG field 131
 MULTIPLE STRINGS field 200
 OPERATION field 318
 PACK DATA field 122
 POOL field 318
 PRINTER CARRIAGE CONTROL field 156
 SEQUENCE NUMBERS field 176
 SPECIFY PACK OPTION FOR "TO" DATA SET field 117
 SRCHFOR process statement 412
 WHEN field 320
 BLKS parameter
 SPACE UNITS field 111
 BLKSZ field 131
 BLKSZ keyword
 SORT command 146
 BLOCK SIZE field 112
 BOTTOM parameter
 BY ROW NUMBER field 298
 COMMAND LINE PLACEMENT field 74
 BREAKPOINT field 324
 Breakpoint Primary Option Menu 323
 Breakpoints (option 7.8)
 control display 322
 defined 319
 finding a breakpoint
 Cancel (option C) 323
 Go (option G) 323
 input errors 322
 line commands
 D (delete) 320
 I (insert) 320
 R (repeat) 320
 primary commands
 CANCEL 320
 END 320, 322
 LOCATE 320
 QUAL 320
 qualification 322
 qualification parameter values 320
 specifying breakpoints 319
 syntax checking 322
 Breakpoints panel 319
 with qualification parameter values 321
 Browse - Entry Panel 85
 Browse (option 1) 85

- Browse commands
 - BROWSE 87
 - COLUMNS 88
 - DISPLAY 89
 - displaying after execution 87
 - FIND 90
 - HEX 95
 - LOCATE 96
 - RESET 97
- Browse data display 86
- browsing DBCS data as EBCDIC data 51
- current position in data set 96
- ending Browse 87
- mixed mode 51
- overview 6
- saving scroll value 27
- selecting Browse 85
- valid logical record length 85
- valid record format 85
- BROWSE command
 - Browse (option 1) 87
 - parameter
 - member 88
- Browse Command Panel 88
- Browse data display 86
- BROWSE field 234
- Browse hexadecimal display
 - data 96
 - vertical 96
- BROWSE Output field 186
- BY ROW NUMBER field 298
- BY VARIABLE field 299
- bypassing menus 5
- BYTE parameter
 - Compare Type field 184

C

- C (catalog data set) line command
 - Data Set List utility (option 3.4) 140
- C (continuation) keyword
 - Search-For utility (option 3.14) 202
- CALL statement format 245
- Canadian French terminals 65
- Cancel (option C) 323
- CANCEL command
 - Breakpoints (option 7.8) 320
 - Command Table utility (option 3.9) 162
 - Dialog Test (option 7) 284
 - Tables (option 7.4) 303, 305
 - Traces (option 7.7) 317, 318
 - Variables (option 7.3) 294
- cataloged partitioned data set 33
- CC parameter
 - DISPLAY command 89
- CHANGED field 40
- Changes (option C)

- char parameter
 - DISPLAY command 89
- character strings, use of 93
- character translation for APL and TEXT
 - keyboards 363
- CHARS parameter
 - FIND command 90
- CHG operation
 - OPERATION field 318
- CHNG parameter
 - LISTING TYPE field 175, 184
- CLASS field 156
- CLEAR key 19
- CLIST
 - attention exits 277
 - error exits 277
- CMD field 288
- CMPBOFS process statement
 - keywords
 - NBTM 395, 399
 - NTOP 395, 399
 - OBTM 395, 399
 - OTOP 395, 399
 - parameters
 - hex-offset 395
 - KYWD 395
 - SuperCE utility (option 3.13) 395
- CMPCOLM process statement
 - Extended Search-For utility (option 3.13.S) 396
 - parameters
 - start-colm 396, 397
 - stop-colm 396, 397
 - SuperCE utility (option 3.13) 396
- CMPCOLMN process statement
 - SuperCE utility (option 3.13) 397
- CMPCOLMO process statement
 - SuperCE utility (option 3.13) 397
- CMPLINE process statement
 - Extended Search-For utility (option 3.13.S) 399
 - parameters
 - 'string' 399
 - KYWD 399
 - start-colm 400
 - stop-colm 400
 - SuperCE utility (option 3.13) 399
- CNPML process option
 - SuperCE utility (option 3.13) 381
- COBOL interactive debug (option 4.10A) 241
 - print output data sets 243
 - processing sequence 241
 - symbolic debug data sets 242
- COBOL Interactive Debug panel 241
- COBOL parameter
 - SEQUENCE NUMBERS field 176
- COBOL process option
 - Extended Search-For utility (option 3.13.S) 379
 - SuperCE utility (option 3.13) 379
- COBOL Structuring Facility panel 333, 334
- COBOL/SF-B (option 9.5)
 - COBOL Structuring Facility background
 - dialog 334

- COBOL/SF-F (option 9.4)
 - COBOL Structuring Facility foreground dialog 333
 - code, defined 9
 - col-1 and col-2 parameters
 - FIND command 90
 - column-identification line 88
 - removing with RESET command 97
 - column limitations 94
 - COLUMNS command
 - Browse (option 1) 88
 - Command (option 6) 275
 - interrupting a TSO command or CLIST 277
 - overview 6
 - restrictions for entering a CLIST 277
 - Session Manager, using 277
 - terminal settings 278
 - TSO commands and CLISTs, entering 276
 - COMMAND DELIMITER field 66
 - command error processing 16
 - COMMAND field 8, 14, 129
 - COMMAND LINE PLACEMENT field 74
 - command name abbreviations 365
 - command prefix (&)
 - displaying commands after execution 87
 - Command Table Editing Panel 160
 - Command Table utility (option 3.9)
 - line commands
 - D (delete) 162
 - I (insert) 162
 - R (repeat) 162
 - primary commands
 - CANCEL 162
 - END 162
 - Command Table Utility panel 159
 - commands
 - default PF key settings 19
 - displaying after execution 87
 - entering 8, 14
 - entering long commands 275
 - entering with PF keys 19
 - error processing 16
 - ISPF 12
 - ISPF/PDF notation 15
 - line 6, 14
 - PDF command 4
 - using an alias 4, 5
 - using option parameter 5
 - primary 6, 14
 - retrieving 27
 - scroll 26
 - stacking with delimiter 16
 - TSO 22
 - commands, Dialog Test 282
 - commands, entering 14
 - command error processing 16
 - COMMAND field 14
 - ISPF/PDF command notation 15
 - PF keys 19
 - Compare Type field 183
 - Batch Pascal/VS compiler (option 5.6) 269
 - Batch PL/I optimizing compiler (option 5.5) 268
 - compress request exit 103, 141
 - concatenation
 - during editing 35
 - during language processing 35
 - COND parameter
 - BROWSE Output field 186
 - CONFIRM command
 - Data Set List utility (option 3.4) 142
 - parameters
 - OFF 142
 - ON 142
 - Confirm Delete panel 114.2
 - CONFIRM DELETE REQUEST field 127
 - Confirm Purge panel 114.2
 - Confirm Table Row Delete panel 301
 - control file 327
 - controlling ISPF libraries with LMF 32
 - library concatenation 34
 - library member change and promotion 36
 - Convert Menus/Messages panel 163
 - copy panel 164
 - Convert Menus/Messages utility (option 3.10)
 - options
 - 1 - convert old format menus to new format panels 164
 - 2 - convert old format messages to new format messages 165
 - COPY command
 - TSO 159
 - Copy Format Selection List panel 169
 - COPY statement 211
 - copying and locking members 119
 - CREATED field 40, 131
 - CREATED keyword
 - SORT command 146
 - Cross System Product/Application Development panel 330
 - Cross System Product/Application Execution panel 331
 - CSP/AD (option 9.1)
 - Cross System Product/Application Development 330
 - CSP/AE (option 9.2)
 - Cross System Product/Application Execution 331
 - CURRENT ROW COUNT field 308, 309
 - CURRENT ROW field 298
 - CURRENT ROW POINTER field 306, 309
 - CURRENT STATUS field 324
 - CURSOR command
 - ISPF 13
 - CURSOR FIELD field 291
 - cursor movement keys 22
 - CURSOR parameter
 - SCROLL field 8
 - CURSOR POSITION field 291
 - CYLS parameter
 - SPACE UNITS field 111

D

- D (DBCS) parameter
 - FIELD TYPE field 167
- D (delete) line command
 - Breakpoints (option 7.8) 320
 - Command Table utility (option 3.9) 162
 - Data Set List utility (option 3.4) 47, 139
 - Dialog Test (option 7) 285
 - Format Specification utility (option 3.11) 170
 - Library utility (option 3.1) 47
 - SCRIPT/VS (option 4.9) 237
 - Tables (option 7.4) 303, 305
 - Traces (option 7.7) 317, 318
 - Variables (option 7.3) 294
- D (delete) parameter
 - Process option field 24, 67
- DA data set organization
 - DSORG field 131
- DASD
 - See direct access storage device
- DATA CLASS field 114.5
- data entry panels 9
- Data Facility Product 108
- DATA parameter
 - CMPLINE process statement 399
 - HEX command 95
 - SCROLL field 8
- data set allocation exit 109, 114.4
- data set control blocks 137
- data set format 10
- Data Set Information panel
 - long format 104
 - long format for managed data sets 105
 - long format for managed data sets, no directory
 - block information 106
 - short format 106
 - short format on managed volumes 107
- Data Set List utility (option 3.4)
 - data set list exit 124
 - data set list listings 347
 - display views and panels
 - ATTRIB 126, 129
 - sequence of display views 126
 - SPACE 126, 128
 - TOTAL 126, 129
 - VOLUME 126, 128
- line commands
 - = (repeat last command) 141
 - B (browse data set) 138
 - C (catalog data set) 140
 - D (delete data set) 139
 - E (edit data set) 138
 - F (free unused space) 141
 - I (data set information) 139
 - M (display member list) 140.1
 - P (print data set) 140
 - R (rename data set) 139
 - S (information (short)) 139
 - U (uncatalog data set) 140
 - X (print index listing) 140
 - Z (compress data set) 141
- line commands that do not support managed data sets
 - C (catalog data set) 140
 - U (uncatalog data set) 140
- line commands that support managed data sets
 - I (data set information) 139
 - R (rename data set) 139
 - S (information (short)) 139
 - X (print index listing) 140
- list data set 143
- member list line commands 47
- options
 - Blank - display data set list 127
 - P - print data set list 135
 - PV - print VTOC information 137
 - V - display VTOC information 136
- primary commands
 - CONFIRM 142
 - FIND 142
 - LOCATE 143
 - RFIND 143
 - SAVE 143
 - SHOWCMD 144
 - SORT 146
- saving scroll value 27
- source and index listings 339
- Data Set List Utility panel 124
- Data Set List Utility SHOWCMD panel 145
- DATA SET NAME field 31, 178, 246
- data set organization 131
- DATA SET PASSWORD field 49
- Data Set utility (option 3.2)
 - option that supports GDG data sets
 - R - rename entire data set 114.1
 - options
 - A - allocate new data set 108
 - Blank - data set information 114.3
 - C - catalog data set 114
 - D - delete entire data set 114.1
 - M - enhanced data set allocation 114.3
 - R - rename entire data set 114
 - S - data set information (short) 114.3
 - U - uncatalog data set 114.1
 - options that do not support managed data sets
 - C - catalog data set 114
 - U - uncatalog data set 114.1
 - options that support managed data sets
 - Blank - data set information 114.3
 - M - enhanced data set allocation 114.3
 - R - rename entire data set 114.1
 - S - data set information (short) 114.3
- DATE CREATED field 307, 309
- DAU data set organization
 - DSORG field 131
- DBCS
 - See double-byte character set
- DBCS COLUMN SPECIFICATION field 299
- DCF

- See Document Composition Facility
- DDDD format
 - EXPIRATION DATE field 113
- DEBUG COMMAND DATA SET field
 - Batch VS COBOL II interactive debug (option 5.10) 272
- DEBUG OPTIONS field
 - Foreground FORTRAN interactive debug (option 4.11) 244
- default mode 7
- default PF key settings 20
 - ending PDF or a PDF function 22
 - getting help 20
 - repeating a FIND or CHANGE command 25
 - scrolling within a data display 25
 - splitting and swapping display screens 21
- DELETE command
 - TSO 127, 128, 142
- delimiters
 - used to stack commands 16
- DELTA parameter
 - LISTING TYPE field 175, 184
- DESCRIPTION column 161
- Description field 238
- DEVICE field 131
- DEVICE keyword
 - SORT command 146
- DFP
 - See Data Facility Product
- Dialog Services (option 7.6)
 - invoking a dialog service 313
- DIALOG SERVICES TO BE TRACED field 316
- Dialog Test (option 7) 279
 - Breakpoints (option 7.8) 319
 - commands 282
 - Dialog Services (option 7.6) 313
 - effect of RETURN command 24
 - environment 280
 - accessing and updating variables 280
 - available ISPF facilities 280
 - exceptions to restoration when using Go (option G) 324
 - ISPF log generation 280
 - error handling, severe 282
 - Exit (option 7.X) 326
 - exiting with jump function (=) 25
 - Functions (option 7.1) 287
 - line commands
 - D (delete lines) 285
 - I (insert lines) 285
 - R (repeat lines) 286
 - Log (option 7.5) 310
 - overview 6
 - Panels (option 7.2) 291
 - primary commands
 - CANCEL 284
 - END 284
 - LOCATE 284
 - QUAL 285
 - RESUME 285
 - message displayed 282
 - reasons errors occur 282
 - Tables (option 7.4) 298
 - Traces (option 7.7) 315
 - Tutorial (option 7.T) 326
 - usual test methods 279
 - variable usage 281
 - Variables (option 7.3) 293
 - why you should generate a log data set 69
- Dialog Test Primary Option Menu 280
- Dialog Test Tutorial panel 326
- dialog, definition of 1
- direct access storage device
 - using with packed data 51
- DIRECTORY BLOCKS field 111
- Display (option 0.4)
 - specifying ASIS 74
 - specifying BOTTOM 74
- Display Characteristics panel 74
- DISPLAY command
 - Browse (option 1) 89
 - parameters
 - CC 89
 - char 89
 - NOCC 89
- display format 7
- Display Panel 291
- Display Row panel 300
- DISPLAY STYLE OPTIONS field 234
- DLMDUP process option
 - SuperCE utility (option 3.13) 381
- DLREFM process option
 - SuperCE utility (option 3.13) 381
- Document Composition Facility 233
- double-byte character set
 - Browse or Edit DBCS data as EBCDIC data 51
 - column specification in dialog test 299
 - defining data type 167
 - Dialog Test variables format 297
 - Edit and Browse 51
 - effect on SuperC utilities 374
 - finding a one-byte hexadecimal string 94
 - format definition 50
 - Format Specification utility (option 3.11) 50, 166
 - formatted data Edit and Browse 50
 - hexadecimal format 51
 - invalid mixed data 51
 - MIXED MODE field 50
 - using the VALUE field 299
- DOWN command
 - ISPF 12, 26
- DPACMT process option
 - Extended Search-For utility (option 3.13.S) 380
 - SuperCE utility (option 3.13) 380
- DPADCM process option
 - Extended Search-For utility (option 3.13.S) 380
 - SuperCE utility (option 3.13) 380
- DPBLKCL process option
 - Extended Search-For utility (option 3.13.S) 380

SuperCE utility (option 3.13) 380
 DPCBCMT process option
 Extended Search-For utility (option 3.13.S) 380
 SuperCE utility (option 3.13) 380
 DPFTCMT process option
 Extended Search-For utility (option 3.13.S) 380
 SuperCE utility (option 3.13) 380
 DPLINE process statement
 Extended Search-For utility (option 3.13.S) 402
 parameters
 'string' 402
 start-colm 402, 407
 stop-colm 402, 407
 SuperCE utility (option 3.13) 402
 DPLINEC process statement
 Extended Search-For utility (option 3.13.S) 402
 SuperCE utility (option 3.13) 402
 DPPLCMT process option
 Extended Search-For utility (option 3.13.S) 380
 SuperCE utility (option 3.13) 380
 DPPSCMT process option
 Extended Search-For utility (option 3.13.S) 380
 SuperCE utility (option 3.13) 380
 DSCBs
 See data set control blocks
 DSLIST exit 124
 DSNAME LEVEL field 124
 DSORG field 131
 DSORG keyword
 SORT command 146
 DS1-DS4 fields 186, 191

E

E (EBCDIC) parameter
 FIELD TYPE field 167
 E (edit data set) line command
 Data Set List utility (option 3.4) 138
 EBCDIC
 See extended binary coded decimal interchange code
 Edit - Entry Panel 99
 Edit (option 2) 99
 editing a data set 99
 editing DBCS data as EBCDIC data 51
 mixed mode 51
 overview 6
 saving scroll value 27
 valid logical record length 99
 valid record format 99
 EDIT command
 Data Set List utility (option 3.4) 47
 END command
 Breakpoints (option 7.8) 320, 322
 Command Table utility (option 3.9) 162
 Dialog Test (option 7) 284
 ISPF 12, 23
 Tables (option 7.4) 301, 303, 305

Traces (option 7.7) 317, 318
 Variables (option 7.3) 294
 ending PDF or a PDF function
 END command 23
 Exit option (X) 25
 RETURN command 24
 ENTER SESSION MANAGER MODE field 277
 ENTER TSO COMMAND OR CLIST BELOW
 field 275
 entering commands with PA and PF keys
 program access (PA) keys 18
 program function (PF) keys 19
 ENVIRON command
 ISPF 13, 81
 environment, Dialog Test 280
 ERASE INPUT key 19
 error handling, Dialog Test 282
 errors
 trigger statement 215
 Example of Using Aspect Ratio Parameter 0 79
 Example of Using Aspect Ratio Parameter 1 80
 examples
 = (repeat last command) line command 141
 account information 56
 additional input libraries 216
 aspect ratio parameter 0 79
 aspect ratio parameter 1 80
 Assembler H allocation data sets
 SYSIN 367
 SYSLIB 367
 SYSLIN 368
 SYSPRINT 368
 SYSPUNCH 369
 SYSTEM 368
 SYSUT1 369
 BROWSE command 88
 bypassing menus 5
 CMPBOFS process statement 395
 CMPCOLM process statement 396
 CMPCOLMN process statement 398
 CMPCOLMO process statement 398
 CMPLINE process statement 400
 column-identification line 89
 command error processing 16
 command interpretation 276
 command table usage restriction 159
 CONFIRM command 142
 Copy Format Selection List panel 169
 data entry panel 85
 data set information displays 104, 105, 106, 107
 data set name levels 124
 default PF key commands 16
 defining PF key functions
 using a BLANK label 73
 using colon (:) to define line commands 72
 Dialog Test
 breakpoint cancel (option C) 279
 changing variable values 295
 DBCS value format 299
 detailed ISPF log entries 311
 dialog trace ISPF log entry 310

DISPLAY service 313
 hexadecimal format 325
 how dialog parts change 324
 ISPF log entry for current value of variable 312
 modifying a table row 302
 NEWPOOL option 281
 no data written to ISPF log 310
 processing that causes a service call 315
 severe error found at breakpoint 282
 specifying breakpoints 321
 why many ISPF log entries can appear 311
 DISPLAY command 90
 display of data set format 86
 displaying commands after execution 87
 DPLINE process statement 403
 DPLINEC process statement 403
 entering generation data set name 32
 F (free unused space) line command 141
 FIND command 142
 FIND command search strings 91
 Foreground Assembler processing 216
 HEX command 95
 INCLUDE statement 232
 invalid load module directory fields 42
 invalid mixed data 51
 ITEXT data set names 227
 JCL generation for compilers 256
 library concatenation 34
 Linkage Editor concatenation sequence 232
 Linkage Editor search sequence 232
 LNCT process statement 404
 LOCATE command 44, 97, 143, 170
 LSTCOLM process statement 405
 member expansion
 code that cannot be processed 212
 member list with expanded line command field 140.1
 member name in title 30
 member parts list 247
 member selection list with statistics 40
 menu 5
 naming a command table 159
 NCHGT process statement 407
 NTITLE process statement 408
 object data set names 218
 OCHGT process statement 407
 panel format 7
 pattern, member selection list 39
 PDF terminal operations 56
 PFSHOW command 73
 PL/I optimizing compiler 268
 print output data set names 243
 printing panels with long lines 77
 printing, deleting, and renaming members 48
 S (select) line command 45
 SAVE command 144
 Search-For list data set name 191
 search-for listing data set name 200
 Search-For process statements 193
 SELECT command 45

SELECT process statement 409
 selecting a tutorial topic 336
 selecting an option 5
 Session Manager control 206
 SHOWCMD command 144
 SORT command 146, 171
 specifying a qualifier with an asterisk 125
 splitting the display screen 21
 SRCHFOR process statement 412
 SRCFORC process statement 412
 stacking commands 16, 66
 substituting '/' for data set name 132
 SuperC list data set name 175, 185
 SuperC process statements 195
 SWAP command 22
 symbolic debug data set names 243
 three-level library hierarchy 33
 typing over parameters 27
 typing over scroll amount 27
 uses for library levels 34
 using GROUPS FOR PRIMARY MEMBERS field 246
 using SuperC 375
 using the Library utility 48
 value in CHANGED field 40
 value in CREATED field 40
 value in LIB field 40
 value in VV.MM field 40
 VOLUME view display 128
 VS COBOL II allocation data sets
 SYSIN 367
 SYSLIB 367
 SYSLIN 368
 SYSPRINT 368
 SYSPUNCH 369
 SYSTEM 368
 SYSUT1 369
 SYSUT2 - SYSUT5 369
 VTOC display 136
 with the TSO/E Information Center Facility 239
 with TSO/E Information Center Facility 218
 327x terminal keys 19
 EXECUTION PARMS field 208, 242
 Foreground PL/I checkout compiler (option 4.4) 226
 Exit (option X)
 bypassing the log and list data set panel 23
 ending PDF from the ISPF/PDF Primary Option Menu 23
 ending PDF on a split screen 22
 overview 7
 similarities to RETURN command 24
 use of log/list defaults 25
 used with jump function (=) 25
 Exit (option 7.X)
 ending Dialog Test 326
 expansion trigger, defined 211
 EXPIRATION DATE field 112
 EXPIRES field 131

EXPIRES keyword

SORT command 146

explosion chain, defined 249

extended binary coded decimal interchange code

Browse or Edit DBCS data as EBCDIC 51

defining data type 167

format definition 50

hexadecimal format 51

invalid mixed data 51

MIXED MODE field 50

using numbered data 168

Extended Search-For - Concatenation Data Set

Entry panel 192

Extended Search-For Compare utility 188

applications 375

options

Blank - search-for strings 191

E - edit Search-For statements data set 193

P - select Search-For process options 192

process options 378

process statements 393

return codes 377

SuperC program description 372

utility differences 371

Extended Search-For process options

ANYC 379

APNDLST 381

COBOL 379

DPACMT 380

DPADCMT 380

DPBLKCL 380

DPCBCMT 380

DPFTCMT 380

DPPLCMT 380

DPPSCMT 380

IDPFX 381

LMTO 381

LONGLN 381

LPSF 382

LTO 382

NOPRTCC 382

NOSEQ 379

SEQ 379

Extended Search-For process statements

* and * (comment lines) 401

CMPCOLM (compare columns) 396

CMPLINE (compare line) 399

DPLINE (do not process line) 402

DPLINEC (continuation of do not process line) 402

LNCT (line count) 404

LSTCOLM (list columns) 405

NCHGT (new change text) 406

notation conventions 393

NTITLE (new title) 408

SELECT (select PDS members) 409

SLIST (include statements in listing) 411

SRCHFOR 412

SRCHFORC 412

Extended Search-For Utility panel 189

F

F (foreground) parameter

MODE field 173

F (free unused space) line command

Data Set List utility (option 3.4) 141

F (function) pool

POOL field 318

F record format

RECORD FORMAT field 111

false match correction notice 356

FBA record format

RECORD FORMAT field 76

FIELD LENGTH field 167

field mark character 19

FIELD NUMBER field 167

FIELD TYPE field 167

field value abbreviations 365

field1 parameter

SORT command 46, 146

field2 parameter

SORT command 46, 146

FILE comparison of a complete PDS 359

FILE comparison of a sequential data set or membered PDS 358

FILE parameter

Compare Type field 183

FIND command

Browse (option 1)

character strings, use of 93

column limitations 94

conditions for character string matches 92

displaying column numbers searched 91

omitting string delimiters 91

one-byte hexadecimal strings 94

picture strings, use of 93

rules for using col-1 and col-2 94

specifying FIND strings 90

starting point, direction, and extent of search 91

string not found actions 92

text strings, use of 92

using RFIND 94

using string delimiters 91

Data Set List utility (option 3.4) 142

parameters

CHARS|PREFIX|SUFFIX|WORD 90

col-1 and col-2 90

NEXT|ALL|FIRST|LAST|PREV 91

string 90

find-mask parameter

NCHGT process statement 406

OCHGT process statement 406

FIRST parameter

FIND command 91

FMVLNS process option

SuperCE utility (option 3.13) 381

FO parameter

ATTRIBUTES field 42
 Foreground (option 4) 205
 Assembler H (option 4.1) 219
 Assembler XF (option 4.1A) 220
 COBOL interactive debug (option 4.10A) 241
 compilation limitations 22
 expanding packed data 211
 FORTRAN interactive debug (option 4.11) 244
 input data sets 215
 Linkage Editor (option 4.7) 231
 list data sets 216
 Member Parts List (option 4.12) 245
 object data sets 217
 OS/VS COBOL compiler (option 4.2A) 223
 output listings 54
 overview 6
 Pascal/VS compiler (option 4.6) 229
 password protection 217
 PL/I checkout compiler (option 4.4) 226
 PL/I optimizing compiler (option 4.5) 228
 processing sequence 205
 SCRIPT/VS (option 4.9) 233
 using the TSO/E Information Center Facility 218
 VS COBOL II compiler (option 4.2) 221
 VS COBOL II interactive debug (option 4.10) 240
 VS FORTRAN compiler (option 4.3) 225
 Foreground Assembler H panel 219
 Foreground Assembler XF panel 220
 Foreground Environment Feature 233
 Foreground Linkage Edit panel 231
 Foreground Member Parts List panel 245
 Foreground OS/VS COBOL Compile panel 223
 Foreground Pascal/VS Compile panel 229
 Foreground PL/I Checkout Compile panel 226
 Foreground PL/I Optimizing Compile panel 228
 Foreground Print Options for Style panel 236
 with the TSO/E Information Center Facility 239
 Foreground Print Options panel 210
 with TSO/E Information Center Facility 218
 Foreground Selection Panel 205
 Foreground VS COBOL II Compile panel 221
 Foreground VS COBOL II Interactive Debug panel 240
 Foreground VS FORTRAN Compile panel 225
 format definition 50
 Format Definition panel 167
 format definitions 50
 FORMAT NAME field 50, 166
 effect on MIXED MODE field 50
 Format Selection List panel 170
 Format Specification panel 166
 Format Specification utility (option 3.11)
 IBM 5550 terminal support 50
 line commands
 D (deleting a format) 170
 R (renaming a format) 171

U (updating a format) 171

options

A - add a new format 167
 C - copy formats 168
 D - delete a format 169
 L or Blank - display format list 169
 U - update a format 169
 primary commands
 LOCATE 170
 SORT 171
 format, panel 7
 formatted data set 50
 FORTRAN interactive debug (option 4.11) 244
 FORTRAN Interactive Debug panel 244
 FREE DSCBS field 137
 FREE EXTENTS field 137
 FREE parameter
 ACTION field 121
 FREE SPACE field 137
 FROM FORMAT field 166
 FROM TABLE field 167
 FSCR parameter
 MODE field 288, 289
 FUNCTION field 316, 318, 320, 324
 parameter
 Function Traces panel 316
 Functions (option 7.1)
 testing a dialog function 287

G

generation data sets 32
 GENERIC UNIT field 110, 114.6
 GENHELP keyword 164
 GET operation
 OPERATION field 318
 Go (option G) 323
 graphics interface mode, effect on Session Manager 207
 GROUP field 30
 GROUPS FOR PRIMARY MEMBERS field 246, 249
 GWCBCL process option
 SuperCE utility (option 3.13) 381

H

HALF parameter
 SCROLL field 8
 Hardcopy utility (option 3.6)
 additional batch printing information 153
 source and index listings 339
 using the TSO/E Information Center Facility 153
 Hardcopy Utility panel

- after JCL generation 152
- before JCL generation 150
- with the TSO/E Information Center Facility
 - installed 154
- hashsum, defined 358
- HELP command
 - displaying column numbers searched 91
 - ISPF 12, 20
 - Tutorial (option T) 338
- HELP reference 165
- Help Tutorial panel 338
- HEX command
 - Browse (option 1) 95
 - parameters
 - DATA 95
 - OFF 95
 - ON 95
 - VERT 95
- hex-offset parameter
 - CMPBOFS process statement 395
- hexadecimal data
 - Dialog Test variables format 296

I

- I (data set information) line command
 - Data Set List utility (option 3.4) 139
- I (implicit) variable
 - P (pool) field 294
- I (insert) line command
 - Breakpoints (option 7.8) 320
 - Command Table utility (option 3.9) 162
 - Dialog Test (option 7) 285
 - Tables (option 7.4) 303, 305
 - Traces (option 7.7) 317, 318
 - Variables (option 7.3) 294
- IBM Products (option 9) 329
 - options
 - 1 - CSP/AD 330
 - 2 - CSP/AE 331
 - 3 - INFO/SYS 332
 - 4 - COBOL/SF-F 333
 - 5 - COBOL/SF-B 334
 - overview 6
- ID field 41, 289
- IDPF process option
 - Extended Search-For utility (option 3.13.S) 381
- IEBCOPY utility 103, 117, 141
- IEBGENER utility 153
- IF PARTITIONED, REPLACE LIKE-NAMED MEMBERS field 117
- IF SEQUENTIAL, "TO" DATA SET DISPOSITION field 117
- INCLUDE statement 211
- index listings
 - index listings for load libraries 342.1
 - managed 342.2
 - index listings for source libraries 341

- managed 342
- index listings for load libraries 342.1
- managed 342.2
- index listings for source libraries 341
- managed 342
- INFO/SYS (option 9.3)
 - Information/System 332
- Information Center Facility 150, 157, 209, 235
- Information/System panel 332
- INIT field 41
- INIT keywords 164
- INITIAL DISPLAY VIEW field 125
- views
 - ATTRIB 126
 - SPACE 126
 - TOTAL 126
 - VOLUME 126
- INITIAL MACRO field 100
- input data sets 215
- INPUT FIELD PAD field 65
- installation exits
 - compress request 103, 141
 - data set allocation 109, 114.4
 - data set list (DSLST) 124
 - print utility 150, 157, 209, 235
 - promote 120
- Interactive System Productivity Facility
 - relationship to PDF 1
- internal character representations for APL
 - keyboards 363
- internal character representations for TEXT
 - keyboards 364
- interpretable text data sets 227
- introduction to PDF
 - commands, entering 14
 - entering commands with PF keys 19
 - interacting with PDF 4
 - invoking PDF 4
 - ISPF/PDF primary options 5
 - remembering information with user profiles 28
 - understanding PDF panels 7
 - using the default PF key settings 20
- Invoke Dialog Function/Selection Menu 287
- Invoke Dialog Service panel 313
- IS data set organization
 - DSORG field 131
- ISPCMD 159
- ISPF
 - See Interactive System Productivity Facility
- ISPF commands
 - assigning 70
 - CURSOR 13
 - DOWN 12, 26
 - END 12, 23
 - ENVIRON 13, 81
 - HELP 12, 20
 - ISPPREP 13
 - jump function (=) 13
 - KEYS 13, 19
 - LEFT 12, 26
 - LIST 13

LOG 13
 PANELID 13
 PFSHOW 13, 72
 PRINT 13
 PRINT-HI 13
 PRINTG 13, 78
 PRINTL 13
 PRINTLHI 13
 RCHANGE 12, 25
 RETRIEVE 12
 RETURN 12, 24
 RFIND 12, 25
 RIGHT 12, 26
 SPLIT 12, 21
 SPLITV 13, 66
 SWAP 12, 22
 taking a screen snapshot 53
 UP 12, 26
 ISPF ENVIRON Command Settings panel 81
 ISPF library names 29
 ISPF library, defined 29
 ISPF log listings 343
 ISPF member statistics 39
 ISPF Parameter Options panel 63
 ISPF Parms (option 0) 63
 Display (option 0.4) 74
 Environ (option 0.7) 81
 Graphic (option 0.6) 78
 List (option 0.5) 76
 Log/List (option 0.2) 67
 overview 6
 PF Keys (option 0.3) 70
 Terminal (option 0.1) 64
 ISPF Tutorial and Introduction panels 336
 ISPF/PDF command notation 15
 ISPF/PDF Primary Option Menu 5
 ISPF/PDF primary options
 option selection 5
 ISPPREP command
 ISPF 13
 ISPTLIB 159
 ISRCFIL control file 327
 ISRFORM table 168
 ISRLEMX 256
 ISRSCAN 256
 ISU data set organization
 DSORG field 131
 ITEXT data sets
 See interpretable text data sets

J

JCL
 See job code language
 JCL generation
 Assembler and Linkage Editor 257
 compilers 256
 job code language

generating and submitting print jobs 151
 job statement information
 parameters
 (ACCOUNT) 56
 /* lines 56
 'NAME' 56
 user-ida 55
 preventing JES line counting 236
 JOB STATEMENT INFORMATION field 24, 68
 JOBID field 156
 JOBNAM field 156
 jump function (=)
 ISPF 13, 25

K

K (keep) parameter
 Process option field 24, 67
 K (key) variable
 T (variable types) field 300, 302, 304
 Katakana terminals 65
 KB attribute
 SPACE UNITS field 114.6
 keys
 ATTENTION (PA1) 18
 CLEAR 19
 cursor movement 22
 ERASE INPUT 19
 ISPF 13
 PA1 277
 RESET 19, 277
 RESHOW (PA2) 19
 KEYS command
 ISPF 19, 70
 KEYS field 306
 keyword parameter
 CMPBOFS process statement 395
 CMPLINE process statement 399
 keyword/operand abbreviations 366
 KN (keep and allocate new data set) parameter
 Process option field 24, 67

L

LABEL fields 72
 label parameter
 LOCATE command 97
 LANG field 288
 LANGUAGE field 246
 LARGEST field 137
 LAST DATE MODIFIED field 307, 309
 LAST MODIFIED BY field 308, 309
 LAST parameter
 FIND command 91
 LAST SERVICE RETURN CODE field 309

- LAST TABLE SERVICE field 309
- LAST TIME MODIFIED field 307, 309
- lcmd parameter
 - SELECT command 45
- LEFT command
 - ISPF 12, 26
- LEL
 - See Linkage Editor language
- LIB field 40, 42
- libraries and data sets
 - data set passwords 49
 - format definitions 50
 - list and log data sets 52
 - member selection lists 37
 - mixed mode 50
 - naming ISPF libraries and data sets 29
 - packed data sets 51
 - sample PDF session 56
 - using LMF to control ISPF libraries 32
- library 33
- library concatenation
 - during editing 35
 - during language processing 35
 - functions that use concatenation 35
 - GROUP field 34
- library controls 32
- Library Management Utilities panel 327
- library member modification and promotion
 - locking members while editing 36
- Library utility (option 3.1)
 - member list line commands 47
 - options
 - B - browse member 103
 - Blank - display member list 102
 - C - compress data set 103
 - D - delete member 104
 - I - data set information 104
 - L - print entire data set 104
 - P - print member 103
 - R - rename member 104
 - S - data set information (short) 106
 - X - print index listing 103
 - options that support managed data sets
 - I - data set information 105
 - S - data set information (short) 107
 - X - print index listing 103
 - source and index listings 339
- Library Utility panel 102
- Line Cmd field 237
- line commands
 - Command Table utility (option 3.9) 162
 - Data Set List utility (option 3.4) 131, 138
 - defined 6, 14
 - Dialog Test (option 7) 285
 - Format Specification utility (option 3.11) 171
 - member selection list 43
 - SCRIPT/VS (option 4.9) 237
- LINE LENGTH field 77
- line-number parameter
 - LOCATE command 97
 - line parameter
- Compare Type field 183
- MODE field 288, 289
- Lines per page field 67
- Linkage Editor (option 4.7) 231
 - concatenation sequence 232
 - LEL control statements 232
- Linkage Editor (option 5.7) 270
- Linkage Editor concatenation sequence 232
- Linkage Editor language 232
- LINKAGE EDITOR OPTIONS field
 - Foreground Linkage Editor (option 4.7) 231
- List (option 0.5)
 - printing panels with long lines 77
- list and log data sets
 - foreground and batch output listings
 - list-id parameter 54
 - prefix parameter 54
 - userid parameter 54
- JOB STATEMENT INFORMATION 55
 - list data set 53
 - log data set 53
 - other temporary data sets 54
 - prefix parameter 52
 - printing list/log data sets 53
 - taking a screen snapshot 53
 - temporary names 52
 - userid parameter 52
- LIST command
 - ISPF 13
- List Data Set Characteristics panel 76
- list data sets 216
 - parameters
 - list-id 44, 143
 - prefix 44, 143
- LIST ID field 208, 216, 253
- list-id parameter
 - Foreground and Batch output listings 54
 - SAVE command 44, 143
- LIST keyword 164
- list of abbreviations 365
 - command names 365
 - field values 365
 - keywords/operands 366
 - programming languages 366
 - scroll amounts 366
- List VTOC utility (option 3.7), removal of 155
- LISTING DS NAME field 175, 191
- Listing Dsn field 184
- LISTING DSNAME field 200
- listing formats
 - data set list listings 347
 - ISPF log listings 343
 - member list listings 344
 - source and index listings 339
 - SuperC listings 348
- LISTING TYPE field 175, 184
- LM Utilities (option 8) 327
 - overview 6
- LMF migration utility 287
- LMTO process option
 - Extended Search-For utility (option 3.13.S) 381

- LNCT process statement
 - Extended Search-For utility (option 3.13.S) 404
 - parameter
 - nnnnnn 404
 - SuperCE utility (option 3.13) 404
- load module library
 - invalid directory fields (?) 42
 - member statistics 41
- load module library display 41
 - expanded line command field 141
- Local printer ID field 24, 67
- LOCATE command
 - Breakpoints (option 7.8) 320
 - Browse (option 1) 96
 - Data Set List utility (option 3.4) 143
 - Dialog Test (option 7) 284
 - Format Specification utility (option 3.11) 170
 - member selection list 43
 - parameters
 - label 97
 - line-number 97
 - lparm 143
 - name 170
 - string 43, 285
 - Tables (option 7.4) 301, 303, 305, 306
 - Traces (option 7.7) 317, 318
 - Variables (option 7.3) 294
- LOCK field 36
- locking members while editing
 - conditions when locking causes errors 37
 - conditions when locking is ignored 37
- locking members with copy/lock 119
- LOCS process option
 - SuperCE utility (option 3.13) 381
- Log (option 7.5)
 - displaying the ISPF log 310
 - logged trace output 310
 - reasons log is not available 310
- log and list data set default options
 - changing primary and secondary page values 69
- Log and List Defaults panel 67
- LOG command
 - ISPF 13
- log data set 53
- Log/List (option 0.2)
 - changing defaults 7
 - effect on Exit (X) option 25
 - log and list data set default options 68
- LOGICAL RECORD LENGTH field 76
- LONG MESSAGE field 8
- LONG parameter
 - LISTING TYPE field 175, 184
- LONGLN process option
 - Extended Search-For utility (option 3.13.S) 381
 - SuperCE utility (option 3.13) 381
- lparm parameter
 - LOCATE command 143
- LPSF process option
 - Extended Search-For utility (option 3.13.S) 382
- LRECL field 131
- Member Parts List (option 5.12) 273
 - 1 - browse/print member parts list 247
 - 2 - write member parts data set 248
- MANAGEMENT CLASS field 114.5
- MAX parameter
 - SCROLL field 8
- MB attribute
 - SPACE UNITS field 114.7
- member expansion
 - block size 212
 - COPY statement 214
 - FIELD A 214
 - FIELD B 214
 - IDENTIFICATION DIVISION 214
 - language processing table 212
 - restrictions 213
 - return codes 215
 - trigger statement errors 215
 - WITH DEBUGGING MODE clause 214
- MEMBER field 30
- member list display 40
 - expanded line command field 140.1
- member list listings
 - load libraries 345
 - source libraries 344
- member list lists 344
- member not found 249
- member parameter
 - BROWSE command 88
 - SELECT process statement 409
 - SYSTEM DD statement 257
- Member Parts List (option 4.12) 245
 - CALL statement format 245
 - member not found 249
 - options
 - 1 - browse/print member parts list 247
 - 2 - write member parts data set 248
 - PROCEDURE DIVISION 245
 - record format 248
- Member Parts List (option 5.12) 273
- MANAGEMENT CLASS field 114.5
- MAX parameter
 - SCROLL field 8
- MB attribute
 - SPACE UNITS field 114.7
- member expansion
 - block size 212
 - COPY statement 214
 - FIELD A 214
 - FIELD B 214
 - IDENTIFICATION DIVISION 214
 - language processing table 212
 - restrictions 213
 - return codes 215
 - trigger statement errors 215
 - WITH DEBUGGING MODE clause 214
- MEMBER field 30
- member list display 40
 - expanded line command field 140.1
- member list listings
 - load libraries 345
 - source libraries 344
- member list lists 344
- member not found 249
- member parameter
 - BROWSE command 88
 - SELECT process statement 409
 - SYSTEM DD statement 257
- Member Parts List (option 4.12) 245
 - CALL statement format 245
 - member not found 249
 - options
 - 1 - browse/print member parts list 247
 - 2 - write member parts data set 248
 - PROCEDURE DIVISION 245
 - record format 248
- Member Parts List (option 5.12) 273
- MANAGEMENT CLASS field 114.5
- MAX parameter
 - SCROLL field 8
- MB attribute
 - SPACE UNITS field 114.7
- member expansion
 - block size 212
 - COPY statement 214
 - FIELD A 214
 - FIELD B 214
 - IDENTIFICATION DIVISION 214
 - language processing table 212
 - restrictions 213
 - return codes 215
 - trigger statement errors 215
 - WITH DEBUGGING MODE clause 214
- MEMBER field 30
- member list display 40
 - expanded line command field 140.1
- member list listings
 - load libraries 345
 - source libraries 344
- member list lists 344
- member not found 249
- member parameter
 - BROWSE command 88
 - SELECT process statement 409
 - SYSTEM DD statement 257
- Member Parts List (option 4.12) 245
 - CALL statement format 245
 - member not found 249
 - options
 - 1 - browse/print member parts list 247
 - 2 - write member parts data set 248
 - PROCEDURE DIVISION 245
 - record format 248
- Member Parts List (option 5.12) 273

member parts list display 247
 member selection lists
 displaying member lists 38
 fields 40
 load module library 42
 source library 40
 ISPF member statistics 39
 line commands
 B (browse) 47
 D (delete) 47
 E (edit) 47
 P (print) 47
 R (rename) 47
 S (select) 45
 list data set 44
 load module library member statistics 41
 primary commands
 LOCATE 43
 RESET 44
 SAVE 44
 SELECT 45
 SORT 46
 saving scroll value 27
 table of differences 38
 updating a member list 49
 membered PDS, defined 174
 menus 9
 Message Display Panel 292
 MESSAGE field 129
 MESSAGE ID field 291
 MESSAGE keyword
 SORT command 146
 messages
 as means of communication 4
 conditions for display 16
 displayed by HELP command 20
 failure to enter required value 9
 inconsistent values 9
 LONG MESSAGE field 8
 SHORT MESSAGE field 8
 mixed data
 assumed 51
 B (browse data set) line command 138
 defining data type 167
 E (edit data set) line command 138
 format definition 50
 invalid, examples of 51
 MIXED MODE field 50
 non-mixed mode 51
 unformatted 51
 using numbered data 168
 MIXED MODE field 50
 MOD field 41
 MOD parameter
 IF SEQUENTIAL, "TO" DATA SET
 DISPOSITION field 117
 MODE field 173, 288, 289
 MODIFIED ROW COUNT field 308, 309
 Modify Row panel 302
 MOUNT authority 32, 111
 move members 23

MOVE parameter
 ACTION field 121
 Move/Copy utility (option 3.3)
 alias entries 123
 options
 C and CP - copying data sets 116
 L and LP - copying and locking data sets 119
 M and MP - moving data sets 118
 P and PP - promoting data sets 120
 source and index listings 339
 with load modules 122
 without load modules 122
 Move/Copy Utility panel 115
 multiple search strings 200
 Multiple Search Strings panel 201
 MULTIPLE STRINGS field 200

N

N (name) variable
 T (variable types) field 300, 302, 304
 N (non-modifiable) attribute
 A (variable attributes) field 294
 NAME field 40, 42, 129
 NAME keyword
 LOCATE command 170
 SORT command 146, 171
 NAME parameter
 JOB STATEMENT INFORMATION 56
 NAMES field 306
 naming ISPF libraries and data sets
 ISPF library names 29
 OTHER PARTITIONED OR SEQUENTIAL
 DATA SET NAMES 31
 volume serials 32
 NARROW process option
 SuperCE utility (option 3.13) 382
 national language support ix
 native mode 7
 NBTM keyword
 CMPBOFS process statement 395
 CMPLINE process statement 399
 NCHGT process statement
 Extended Search-For utility (option 3.13.S) 406
 parameters
 'find-mask' 406
 'string' 406
 SuperCE utility (option 3.13) 406
 NEVER parameter
 LOCK field 36
 New DS Name field 182
 new-member parameter
 SELECT process statement 409
 NEW OUTPUT CLASS field 156
 new-title-text parameter
 NTITLE process statement 408
 OTITLE process statement 408

NEW USERID field 147
 NEW VERSION NUMBER field 147
 NEWAPPL field 289
 NEWNAME field 102
 NEWPOOL field 289
 NEXT parameter
 FIND command 91
 nnnnn parameter
 LNCT process statement 404
 NO parameter
 ACTIVE field 316, 318, 320
 BROWSE Output field 186
 CONFIRM DELETE REQUEST field 127
 IF PARTITIONED, REPLACE LIKE-NAMED MEMBERS field 117
 LOCK field 36
 MIXED MODE field 51
 MULTIPLE STRINGS field 200
 NEWAPPL field 289
 NEWPOOL field 289
 PACK DATA field 122
 PASSLIB field 290
 PURGE DATA SET field 114.2
 SPECIFY PACK OPTION FOR "TO" DATA SET field 117
 TABLE AVAILABLE field 307
 TABLE ON DISK field 309
 NOCC parameter
 DISPLAY command 89
 NOLIST parameter
 LISTING TYPE field 175, 184
 non-managed data set 114.3
 non-paired change, defined 356
 NOP action
 ACTION column 161
 NOPRTCC process option
 Extended Search-For utility (option 3.13.S) 382
 SuperCE utility (option 3.13) 382
 NOSEQ parameter
 SEQUENCE NUMBERS field 176
 NOSEQ process option
 Extended Search-For utility (option 3.13.S) 379
 SuperCE utility (option 3.13) 379
 NOT OPEN parameter
 STATUS FOR THIS SCREEN field 307, 308
 NOWRITE parameter
 OPEN OPTION field 309
 NTITLE process statement
 Extended Search-For utility (option 3.13.S) 408
 parameter
 new-title-text 408
 SuperCE utility (option 3.13) 408
 NTOP keyword
 CMPBOFS process statement 395
 CMPLINE process statement 399
 NUMBER OF COPIES field 154
 NUMBER OF KEYS field 306
 NUMBER OF NAMES field 306
 NUMBER OF PF KEYS field 65
 NUMBER OF ROWS field 306

ATTRIBUTES field 42

O

object data sets 217
 OBJECT field
 Batch PL/I checkout compiler (option 5.4) 267
 Foreground PL/I checkout compiler (option 4.4) 226
 Foreground VS FORTRAN compiler (option 4.3) 225
 OBTM keyword
 CMPBOFS process statement 395
 CMPLINE process statement 399
 OCHGT process statement
 SuperCE utility (option 3.13) 406
 OFF parameter
 CONFIRM command 142
 HEX command 95
 SHOWCMD command 144
 OL parameter
 ATTRIBUTES field 42
 Old DS Name field 182
 old-member parameter
 SELECT process statement 409
 OLD parameter
 IF SEQUENTIAL, "TO" DATA SET DISPOSITION field 117
 ON parameter
 CONFIRM command 142
 HEX command 95
 SHOWCMD command 144
 OPEN OPTION field 309
 OPEN parameter
 STATUS FOR THIS SCREEN field 307, 308
 operand notation
 one required ({ }) 15
 optional ([]) 15
 OR symbol (|) 15
 OPERATION field 318
 OPT field 288
 OPTION field 8
 entering commands 14
 option selection 5
 C - Changes 7
 T - Tutorial 7
 X - Exit 7
 0 - ISPFParms 6
 1 - Browse 6
 2 - Edit 6
 3 - Utilities 6
 4 - Foreground 6
 5 - Batch 6
 6 - Command 6
 7 - Dialog Test 6
 8 - LM Utilities 6
 9 - IBM Products 6
 ORIGINAL ROW COUNT field 308, 309

OS/VS COBOL compiler (option 4.2A) 223
 OS/VS COBOL compiler (option 5.2A) 264
 OTHER field

Batch Assembler H (option 5.1) 258
 Batch Assembler XF (option 5.1A) 260
 Batch Linkage Editor (option 5.7) 270
 Batch OS/VS COBOL compiler (option 5.2A) 264
 Batch PL/I checkout compiler (option 5.4) 267
 Batch VS COBOL II compiler (option 5.2) 262
 Batch VS FORTRAN compiler (option 5.3) 265
 Foreground OS/VS COBOL compiler (option 4.2A) 223
 Foreground Pascal/VS compiler (option 4.6) 229
 Foreground PL/I checkout compiler (option 4.4) 226
 Foreground PL/I optimizing compiler (option 4.5) 228
 Foreground VS COBOL II compiler (option 4.2) 221
 Foreground VS FORTRAN compiler (option 4.3) 225

OTHER PARTITIONED OR SEQUENTIAL DATA
 SET NAMES 31

other temporary data sets 54

OTITLE process statement

SuperCE utility (option 3.13) 408

OTOP keyword

CMPBOFS process statement 395

CMPLINE process statement 399

OUTDD DD field 178

Outlist utility (option 3.8)

options

Blank - display job output 158

D - delete job output from SYSOUT hold
 queue 157

L - list job names/IDs via the TSO STATUS
 command 157

P - print job output and delete from SYSOUT
 hold queue 157

R - requeue job output to a new output
 class 158

source and index listings 339

Outlist Utility panel 156

OV parameter

ATTRIBUTES field 42

OVSUM listing type

FILE comparison of a complete PDS 359

FILE comparison of a sequential data set or
 membered PDS 358

OVSUM parameter

LISTING TYPE field 175, 184

P

P (pool) field 294

P (prefix) qualifier

SRCHFOR process statement 412

P (print) line command

Data Set List utility (option 3.4) 47, 140

Library utility (option 3.1) 47

P (profile) pool

POOL field 318

P (profile) variable

P (pool) field 294

PACK DATA field 121

packed data sets 51

requirements for using 52

packing data

Foreground (option 4) 206

PAGE parameter

SCROLL field 8

paired change, defined 356

PANEL field 287

panel format 7

PANEL NAME field 291

panel types 8

data entry panels 9

menus 9

scrollable data displays 9

PANELID command

ISPF 13

panels

as means of communication 4

basic types 8

data set list display view panels

ATTRIB view panel 129

SPACE view panel 128

TOTAL view panel 129

VOLUME view panel 128

default mode 7

defined 7

load module library display 41

expanded line command field 141

member list display 40

expanded line command field 140.1

menus 9

code, blank 9

code, defined 9

native mode 7

panel format 7

retaining previous values 9

scrollable format 9

split screen example 21

what they display 4

Panels (option 7.2)

testing dialog panels 291

PARM field 289

Pascal/VS compiler (option 4.6) 229

Pascal/VS compiler (option 5.6) 269

PASSLIB field 290

PASSTHRU action

ACTION column 161
 PASSWORD field 208, 217
 password protection 217
 pattern parameter
 SELECT command 45
 pattern, defined 39
 PA1 (ATTENTION) key 18, 277
 PA2 (RESHOW) key 19
 PCF
 See Programming Control Facility
 PD (print and delete) parameter
 Process option field 24, 67
 PDF
 See Program Development Facility
 PDS Member List field 183
 parameters 189
 PF Key Definitions and Labels - Alternate Keys
 panel 71
 PF Key Definitions and Labels - Primary Keys
 panel 71
 PF Key Definitions and Labels panel 70
 PF Keys (option 0.3)
 assigning ISPF/PDF commands 19, 70
 assigning labels 72
 informing ISPF of the number of PF keys 65
 terminals with 12 PF keys 70
 terminals with 24 PF keys 70
 PF keys, default assignments 11
 PFSHOW command
 ISPF 13, 72
 PGM field 289
 picture strings, use of 93
 PL/I checkout compiler (option 4.4) 226
 PL/I checkout compiler (option 5.4) 267
 PL/I optimizing compiler (option 4.5) 228
 PL/I optimizing compiler (option 5.5) 268
 PO data set organization
 DSORG field 131
 POOL field 318
 POU data set organization
 DSORG field 131
 prefix parameter
 FIND command 90
 Foreground and Batch output listings 54
 list and log data sets 52
 member list list data set 44, 143
 SYSTEM DD statement 257
 PREV parameter
 FIND command 91
 primary commands
 Browse (option 1) 87
 Command Table utility (option 3.9) 162
 Data Set List utility (option 3.4) 142
 defined 6, 14
 Format Specification utility (option 3.11) 170
 member selection list 43
 primary member, defined 249
 Primary pages field 68
 changing values 69
 PRIMARY QUANTITY field 111
 PRINT command

ISPF 13
 Print Graphics Parameters panel 78
 PRINT-HI command
 ISPF 13
 PRINT ID field 242, 243
 PRINT MODE field 236
 print output data sets 243
 print utility exit 150, 157, 209, 235
 PRINTER CARRIAGE CONTROL field 156
 PRINTER FORMAT field 154
 PRINTER LOCATION field 153
 PRINTG command
 ISPF 13, 78
 printing
 list/log data sets 53
 panels with long lines 77
 printing a SuperCE listing in batch mode 188
 printing your output 150
 PRINTL command
 ISPF 13
 PRINTLHI command
 ISPF 13
 PROCEDURE DIVISION 245
 Process option field 24, 67
 Process Options field 185, 191
 PROFILE DSN field 172
 PROFILE NAME field 100
 profiles
 user 28
 PROG ID field 242
 program access (PA) keys 18
 Program Development Facility
 interacting with 4
 introduction to 1
 invoking 4
 program function (PF) keys 19
 Programming Control Facility 49
 programming language abbreviations 366
 project administrator 327
 PROJECT field 29
 promote exits 120
 promote members 35, 36
 PS data set organization
 DSORG field 131
 PSU data set organization
 DSORG field 131
 PURGE DATA SET field 114.2
 PUT operation
 OPERATION field 318



QUAL command
 Breakpoints (option 7.8) 320
 Dialog Test (option 7) 285
 QUAL field 320
 QUALIFICATION PARAMETER VALUES
 field 321

qualifier parameter

SRCHFOR process statement 412

R

R (rename) line command

Data Set List utility (option 3.4) 47, 139

Format Specification utility (option 3.11) 171

Library utility (option 3.1) 47

R (repeat) line command

Breakpoints (option 7.8) 320

Command Table utility (option 3.9) 162

Dialog Test (option 7) 286

Tables (option 7.4) 303, 305

Traces (option 7.7) 317, 318

RACF

See Resource Access Control Facility

RCHANGE command

ISPF 12, 25

REASON CODE field 121

RECFM field 131

RECFM keyword

SORT command 146

RECORD FORMAT field 76, 111

RECORD LENGTH field 112

REFERRED field 131

REFERRED keyword

SORT command 146

REFMOVR process option

SuperCE utility (option 3.13) 382

Rename Data Set panel 114

repeating a FIND or CHANGE command

using the RCHANGE command 25

using the RFIND command 25

RESET command

Browse (option 1) 97

member selection list 44

Reset ISPF Statistics panel 147

Reset ISPF Statistics utility (option 3.5)

options

D - delete ISPF statistics 149

R - reset (create/update) ISPF statistics 148

results of resetting statistics 149

valid logical record lengths 147

RESET key 19, 277

RESET MOD LEVEL field 147

RESET SEQ NUMBERS field 148

RESHOW (PA2) key 19

Resource Access Control Facility 49

restrictions on member expansion and member parts lists

all programming languages 213

Assembler 213

COBOL 214

FORTRAN 214

Pascal 214

PL/I 214

SCRIPT/VS 214

RESUME command

Dialog Test (option 7) 285

RETRIEVE command

ISPF 12

retrieving commands 27

RETURN CODE field 325

return codes

Batch JCL generation 256

member expansion 215

SuperC 377

RETURN command

ISPF 12, 24

RETURN keyword 164

RF parameter

ATTRIBUTES field 42

RFIND command

Data Set List utility (option 3.4) 143

ISPF 12, 25

RIGHT command

ISPF 12, 26

RN parameter

ATTRIBUTES field 42

Rnn parameter

WHEN field 320

ROW IDENTIFICATION field 298

ROW NUMBER field 301

RU parameter

ATTRIBUTES field 42

rules for:

printing panels with long lines 77

service call image for function trace entries 311

specifying log and list options 68

substituting '/' for data set name 132

typing over multiple-choice parameters 27

typing over scroll amounts 27

using the add row option 305

using the col-1 and col-2 operands 94

using the Dialog Test Breakpoints option 322

using the Modify Row option 302

S

S (information (short)) line command

Data Set List utility (option 3.4) 139

S (save) variable

T (variable types) field 300, 302, 304

S (select) line command

Format Specification utility (option 3.11) 171

member selection list 45

SCRIPT/VS (option 4.9) 237

S (shared) pool

POOL field 318

S (shared) variable

P (pool) field 294

S (suffix) qualifier

SRCHFOR process statement 412

S record format

RECORD FORMAT field 112

- sample PDF session 56
- SAVE command
 - Data Set List utility (option 3.4) 143
 - member list listings, examples of 344
 - member selection list 44
 - parameters
 - list-id 44, 143
- SC parameter
 - ATTRIBUTES field 42
- Screen Containing Graphics to be Printed 79
- SCREEN FORMAT field 66
- SCRIPT/VS (option 4.9) 233
 - changing style options 238
 - line commands
 - D (delete) 237
 - S (select) 237
 - processing sequence 233
 - selecting a formatting style 237
 - using the TSO/E Information Center Facility 239
- SCRIPT/VS Options for Style panel 238
- SCRIPT/VS Processor panel 233
- scroll amount abbreviations 366
- scroll amounts 26
- scroll commands 26
- SCROLL field 8
- scrollable data displays 9
 - data set format 10
 - table format 9
- scrolling within a data set
 - typing over scroll and multiple-choice parameters 26
 - using the scroll commands 26
- Search DS Name field 189
- Search-For Process Statements panel 193
- Search-For utility (option 3.14)
 - applications 375
 - return codes 377
 - selecting members from a member list 202
 - SuperC program description 372
 - utility differences 371
- Search-For Utility panel 199
- Search Member List panel 203
- SEARCH STRING field 199
- search strings, multiple 200
- Secondary pages field 68
 - changing values 69
- SECONDARY QUANTITY field 111
- SELECT action
 - ACTION column 160
- SELECT command
 - Format Specification utility (option 3.11) 171
 - member selection list 45
 - parameters
- SELECT process statement
 - Extended Search-For utility (option 3.13.S) 409
 - parameters
 - member 409
 - new-member 409
 - old-member 409
- Select SCRIPT/VS Formatting Style panel 237
- SELECT service 163, 280
- SEQ parameter
 - SEQUENCE NUMBERS field 176
- SEQ process option
 - Extended Search-For utility (option 3.13.S) 379
 - SuperCE utility (option 3.13) 379
- SEQUENCE NUMBERS field 176
- SERVICE field 319
- Service Name field 324
- Session Manager
 - Command (option 6) 277
 - Foreground (option 4) 206
 - Member Parts List (option 4.12) 246
 - SCRIPT/VS 235
- Session Manager, using 277
- SETVERB action
 - ACTION column 161
- shift-in characters 50
 - DBCS character strings 51
 - Dialog Test variables 297
 - formatted data Edit and Browse 50
 - invalid mixed data, examples of 51
 - treatment in non-mixed mode 51
 - using the VALUE field 299
- shift-out characters 50
 - DBCS character strings 51
 - Dialog Test variables 297
 - formatted data Edit and Browse 50
 - invalid mixed data, examples of 51
 - treatment in non-mixed mode 51
 - using the VALUE field 299
- SHORT MESSAGE field 8
- SHOWCMD command
 - Data Set List utility (option 3.4) 144
 - parameters
 - OFF 144
 - ON 144
- SHR NOWRITE parameter
 - OPEN OPTION field 309
- SHR WRITE parameter
 - OPEN OPTION field 309
- SI characters
 - See shift-in characters
- single-byte character set
 - See extended binary coded decimal interchange code
- SIZE field 40, 42, 137
- SLIST process statement
 - Extended Search-For utility (option 3.13.S) 411
 - SuperCE utility (option 3.13) 411
- snapshot of a screen 53
- SO characters
 - See shift-out characters
- SORT command
 - Data Set List utility (option 3.4) 146
 - Format Specification utility (option 3.11) 171
 - keywords
 - %USED 146
 - BLKSZ 146

- CREATED 146
- DEVICE 146
- DSORG 146
- EXPIRES 146
- LRECL 146
- MESSAGE 146
- NAME 146
- RECFM 146
- REFERRED 146
- TRACKS 146
- VOLUME 146
- XT 146
- member selection list 46
- parameters
 - field1 46, 146
 - field2 46, 146
 - NAME 171
 - TIME 171
- source and index listings
 - index listings 341
 - source listings 340
- source data for SuperC listings 349
- SOURCE DATA ONLINE field 252
- SOURCE DATA PACKED field 206, 211, 252
- SOURCE field 242
- source listings 340
- source segments 35
- SOURCE TYPE field 208
- SPACE display view
 - INITIAL DISPLAY VIEW field 126
- SPACE UNITS field 111, 114.6
- Special Display Panel 314
- Specify Disposition of Log and List Data Sets
 - panel 23
- SPECIFY PACK OPTION FOR "TO" DATA SET
 - field 117
- SPF
 - See System Productivity Facility
- SPFUTIL program 163
- SPLIT command 21
 - ISPF 12, 21
 - repositioning a split screen 21
 - split screen limitations 21
- splitting and swapping display screens
 - using the SPLIT command 21
 - using the SWAP command 22
- SPLITV command
 - ISPF 13, 66
- SRCHFOR process option
 - Extended Search-For utility (option 3.13.S) 412
 - parameters
 - qualifier 412
 - string 412
 - qualifiers
 - blank 412
 - P (prefix) 412
 - S (suffix) 412
 - W (word) 412
- SRCHFORC process option
 - Extended Search-For utility (option 3.13.S) 412
- standard Search-For utility
 - entering multiple search strings 200
- start-colm parameter
 - CMPCOLM process statement 396
 - CMPCOLMN process statement 397
 - CMPCOLMO process statement 397
 - CMPLINE process statement 400
 - DPLINE process statement 402
 - DPLINEC process statement 402
 - LSTCOLM process statement 405
 - NCHGT process statement 407
 - OCHGT process statement 407
- START COLUMN field 167
- Statements Dsn field 185, 191
- statistics
 - ISPF members 39
 - load module libraries 41
- statistics, creating or resetting 148
- Status for Table panel
 - table not open 307
 - table open 308
- STATUS FOR THIS SCREEN field 307, 308
- stop-colm parameter
 - CMPCOLM process statement 396
 - CMPCOLMN process statement 397
 - CMPCOLMO process statement 397
 - CMPLINE process statement 400
 - DPLINE process statement 402
 - DPLINEC process statement 402
 - LSTCOLM process statement 405
 - NCHGT process statement 407
 - OCHGT process statement 407
- STORAGE CLASS field 114.5
- Storage Management Subsystem 108, 114.3
 - allocation errors 114.7
 - extra attributes for SPACE UNITS field 114.6
- string parameter
 - DPLINE process statement 402
 - DPLINEC process statement 402
 - FIND command 90
 - LOCATE command 43, 285
 - NCHGT process statement 406
 - OCHGT process statement 406
 - SRCHFOR process statement 412
- STYLE field 233, 238
- SUFFIX parameter
 - FIND command 90
- SuperC Activate/Create Profile options
 - A - activate 197
 - C - create 197
- SuperC compare types
 - BYTE 184
 - FILE 183
 - LINE 183
 - WORD 183
- SuperC listing source data 349
- SuperC listing types
 - CHNG 175, 184
 - DELTA 175, 184
 - LONG 175, 184
 - NOLIST 175, 184

OVSUM 175, 184
 SuperC Member List panel 177
 SuperC process options
 ANYC 379
 APNDLST 381
 APNDUPD 383
 CNPML 381
 COBOL 379
 DLMDUP 381
 DLREFM 381
 DPACMT 380
 DPADCMT 380
 DPBLKCL 380
 DPCBCMT 380
 DPFTCMT 380
 DPPLCMT 380
 DPPSCMT 380
 FMVLNS 381
 GWCBL 381
 LOCS 381
 LONGLN 381
 NARROW 382
 NOPRTCC 382
 NOSEQ 379
 REFMOVR 382
 SEQ 379
 UPDCMS8 383
 UPDCNTL 384
 UPDMVS8 387
 UPDPDEL 388
 UPDSEQ0 389
 UPDSUMO 389
 WIDE 382
 XWDCMP 382
 SuperC Process Statements panel 195
 SuperC Utility - Submit Batch Jobs panel 178
 SuperC Utility - Submit Batch Jobs panel using
 NOLIST 181
 SuperC utility (option 3.12)
 "new" data set 172
 "old" data set 174
 applications 375
 printing a SuperC listing in batch mode 177
 process options 378
 process statements 393
 return codes 377
 specifying members on a member list 176
 SuperC program description 372
 SuperC submit options 179
 blank - generate output listing to SYSOUT
 CLASS 179
 1 - generate output listing in DATA SET
 NAME 179
 2 - generate output listing using completed
 //OUTDD DD 180
 utility differences 371
 SuperC Utility panel 172
 SuperCE - Concatenation Batch Entry panel 188
 SuperCE - Concatenation Interactive Entry
 panel 187
 SuperCE Profile Manager panel 186

SuperCE process statements
 * and * (comment lines) 401
 CMPBOFS (compare byte offsets) 395
 CMPCOLM (compare columns) 396
 CMPCOLMN (compare column new) 397
 CMPCOLMO (compare column old) 397
 CMPLINE (compare line) 399
 DPLINE (do not process line) 402
 DPLINEC (continuation of do not process
 line) 402
 LNCT (line count) 404
 LSTCOLM (list columns) 405
 NCHGT (new change text) 406
 notation conventions 393
 NTITLE (new title) 408
 OCHGT (old change text) 406
 OTITLE (old title) 408
 SELECT (select PDS members) 409
 SLIST (include statements in listing) 411
 SuperCE Profile Browse Display panel 196
 SuperCE utility (option 3.13)
 applications 375
 options
 A - activate/create profiles 195
 B - submit batch data set compare 187
 Blank - compare data sets 186
 E - edit SuperCE statements data set 194
 P - select process options 378
 P - select SuperCE process options 194
 S - Extended Search-For Compare utility 188
 process options 378
 process statements 393
 return codes 377
 SuperC program description 372
 SuperCE utility 182
 utility differences 371
 SuperCE Utility panel 182
 SWAP command
 ISPF 12, 22
 symbolic debug data sets 242
 synchronization data 374
 SYSIN data set
 Assembler H 367
 VS COBOL II 367
 SYSLIB data set
 Assembler H 367
 VS COBOL II 367
 SYSLIN data set
 Assembler H 368
 VS COBOL II 368
 SYSOUT class field 24, 67, 178, 253
 SYSPRINT data set
 Assembler H 368
 VS COBOL II 368
 SYSPUNCH data set
 Assembler H 369
 VS COBOL II 369
 system command table 159
 System Productivity Facility 163
 SYSTEM data set

Assembler H 368
 VS COBOL II 368
 SYSTERM DD statement
 parameters
 member 257
 prefix 257
 SYSUT1 data set
 Assembler H 369
 VS COBOL II 369
 SYSUT2 - SYSUT5 data sets
 VS COBOL II 369

T

T (truncated) attribute
 A (variable attributes) field 294, 300, 302
 T (truncation) column 160
 T (variable types) field 300, 302, 304
 T record format
 RECORD FORMAT field 112
 TABLE AVAILABLE field 307
 table format 9
 table input library 159
 TABLE NAME field 298, 301
 Table of Contents panel 337
 TABLE ON DISK field 309
 Table Structure panel 306
 Tables (option 7.4)
 Add Row line commands
 D (delete) 305
 I (insert) 305
 R (repeat) 305
 Add Row primary commands
 CANCEL 305
 END 305
 LOCATE 305
 Display Row commands
 END 301
 LOCATE 301
 Display Structure command
 LOCATE 306
 Modify Row line commands
 D (delete) 303
 I (insert) 303
 R (repeat) 303
 Modify Row primary commands
 CANCEL 303
 END 303
 LOCATE 303
 options
 1 - display row 300
 2 - delete row 301
 3 - modify row 301
 4 - add row 304
 5 - display structure 306
 6 - display status 307
 working with dialog tables 298
 Tables panel 298

Target Panel for Promoting 121
 temporary data sets 54
 TERM field
 TERMIN command procedure statements 277
 Terminal (option 0.1)
 changing default delimiter 16
 informing ISPF of the number of PF keys 65
 overriding mode switching 7
 Terminal Characteristics panel 64
 TERMINAL TYPE field 64
 terminals
 Batch Assembler H (option 5.1) 258
 Batch Assembler XF (option 5.1A) 260
 Batch Linkage Editor (option 5.7) 270
 Batch VS COBOL II compiler (option 5.2) 262
 Batch VS FORTRAN compiler (option 5.3) 265
 12 PF keys 70
 24 PF keys 70
 TEST command
 TSO 279
 TEST field
 Batch OS/VS COBOL compiler (option 5.2A) 264
 Foreground OS/VS COBOL compiler (option 4.2A) 223
 Foreground VS COBOL II compiler (option 4.2) 221
 testing dialogs 279
 text strings, use of 92
 TIME CREATED field 307, 309
 TIME parameter
 SORT command 171
 TITLE field 8
 To Panel for Copying 116
 To Panel for Moving 118
 TOP parameter
 BY ROW NUMBER field 298
 TOTAL display view
 INITIAL DISPLAY VIEW field 126
 trace output in ISPF log
 function trace entries 311
 trace header entries 310
 variable trace entries 311
 Traces (option 7.7)
 Function Traces line commands
 D (delete) 317
 I (insert) 317
 R (repeat) 317
 Function Traces primary commands
 CANCEL 317
 END 317
 LOCATE 317
 options
 1 - function traces 315
 2 - variable traces 317
 Variable Traces line commands
 D (delete) 318
 I (insert) 318
 R (repeat) 318
 Variable Traces primary commands
 CANCEL 318

- END 318
- LOCATE 318
 - working with trace specifications 315
- Traces panel 315
- TRACKS field 130, 137
- TRACKS keyword
 - SORT command 146
- transmission codes 363
- trigger statement errors 215
- TRKS parameter
 - SPACE UNITS field 111
- TRKS/CYLS field 137
- TS parameter
 - ATTRIBUTES field 42
- TSO Command Processor panel 275
- TSO commands 22
 - ACCOUNT 32, 111
 - ALLOCATE 109, 114.5
 - COPY 159
 - DELETE 127, 128, 142
 - TEST 279
- TSO commands and CLISTs
 - Data Set List utility (option 3.4) 132
 - variables
 - Data Set List utility (option 3.4) 134
 - ZDLBLKSZ 134
 - ZDLCAT 135
 - ZDLCDATE 134
 - ZDLCONF 134
 - ZDLDEV 134
 - ZDLDSN 134
 - ZDLDSORG 134
 - ZDLEDATE 134
 - ZDLEXT 134
 - ZDLLCMD 134
 - ZDLLRECL 134
 - ZDLMSG 135
 - ZDLNDSN 135
 - ZDLRDATE 134
 - ZDLRECFM 134
 - ZDLREF 135
 - ZDLsize 134
 - ZDLUSED 134
 - ZDLVOL 134
- TSO commands and CLISTs, entering 276
- TTR data 104
- TTR field 42
- Tutorial (option T) 335
 - finding ISPF information 4
 - HELP command 338
 - selecting 335
 - selecting the tutorial
 - ending the Tutorial 337
 - selecting a tutorial topic 335
- Tutorial (option 7.T)
 - displaying tutorial information 326
- TYPE field 30
- typing over scroll and multiple-choice parameters 26

U

- U (uncatalog data set) line command
 - Data Set List utility (option 3.4) 140
- U (updating a format) line command
 - Format Specification utility (option 3.11) 171
- U record format
 - RECORD FORMAT field 111
- understanding PDF panels 7
 - panel format 7
 - panel types 8
- UNIT field 137
- unmovable data sets 131
- UP command
 - ISPF 12, 26
- UPD parameter
 - BROWSE Output field 186
- UPDATE COUNT field 308, 309
- UPDATE DS NAME field 174
- Update Dsn field 185
- updating a member list 49
- UPDCMS8 process option
 - SuperCE utility (option 3.13) 383
- UPDCMS8 update data set 383
- UPDCNTL process option
 - SuperCE utility (option 3.13) 384
- UPDCNTL update data sets and formats 384
- UPDMVS8 process option
 - SuperCE utility (option 3.13) 387
- UPDPDEL process option
 - SuperCE utility (option 3.13) 388
- UPDPDEL update data set 388
- UPDSEQ0 process option
 - SuperCE utility (option 3.13) 389
- UPDSEQ0 update data set 389
- UPDSUMO process option
 - SuperCE utility (option 3.13) 389
- UPDSUMO update data set 389
- user-ida parameter
 - JOB STATEMENT INFORMATION 55
- user profiles 28
- userid parameter
 - Foreground and Batch output listings 54
 - list and log data sets 52
- Utilities (option 3) 101
 - Command Table utility (option 3.9) 159
 - Convert Menus/Messages utility (option 3.10) 163
 - Data Set List utility (option 3.4) 124
 - Data Set utility (option 3.2) 108
 - Format Specification utility (option 3.11) 166
 - Hardcopy utility (option 3.6) 150
 - Library utility (option 3.1) 102
 - List VTOC utility (option 3.7), removal of 155
 - Move/Copy utility (option 3.3) 115
 - Outlist utility (option 3.8) 156
 - overview 6
 - Reset ISPF Statistics utility (option 3.5) 147
 - Search-For utility (option 3.14) 199

SuperC utility (option 3.12) 172
 SuperCE utility (option 3.13) 182
 Utility Selection Menu 101

V

V (VDEFINE) variable
 P (pool) field 294
 V record format
 RECORD FORMAT field 111
 VALUE field 294, 299, 300, 302, 304
 specifying a DBCS value 299
 VARIABLE field 293, 300, 302, 304, 317
 Variable Traces panel 317
 variable usage, Dialog Test 281
 Variables (option 7.3)
 line commands
 D (delete) 294
 I (insert) 294
 manipulating variables
 creating new variables 295
 deleting variables 295
 primary commands
 CANCEL 294
 END 294
 LOCATE 294
 usage notes
 DBCS data 297
 hexadecimal data 296
 input errors 295
 split-screen mode 296
 test mode 296
 variable life 296
 variable value 296
 working with dialog variables 293
 Variables panel 293
 VBA record format
 RECORD FORMAT field 76
 VERB column 160
 VERT parameter
 HEX command 95
 VIO, allocating temporary data sets 54
 VOLUME DATA field 137
 VOLUME display view
 INITIAL DISPLAY VIEW field 126
 VOLUME field 125, 130
 VOLUME keyword
 SORT command 146
 VOLUME SERIAL field 32, 110, 114.6
 VS COBOL II compiler (option 4.2) 221
 allocation data sets
 SYSIN 367
 SYSLIB 367
 SYSLIN 368
 SYSPRINT 368
 SYSPUNCH 369
 SYSTEM 368

SYSUT2 - SYSUT5 369
 VS COBOL II compiler (option 5.2) 262
 VS COBOL II interactive debug (option 4.10) 240
 VS COBOL II interactive debug (option 5.10) 272
 VS COBOL II Interactive Debug panel 272
 VS data set organization
 DSORG field 131
 VS FORTRAN compiler (option 4.3) 225
 VS FORTRAN compiler (option 5.3) 265
 VTOC DATA field 137
 VTOC information display 136
 VV.MM field 40

W

W (word) qualifier
 SRCHFOR process statement 412
 WHEN field 320
 WIDE process option
 SuperCE utility (option 3.13) 382
 window as scrolling concept 25
 WORD parameter
 Compare Type field 183
 FIND command 90
 WRITE parameter
 OPEN OPTION field 309
 writing a list to a list data set
 data set list 143
 member list 44

X

X (print index listing) line command
 Data Set List utility (option 3.4) 140
 XT field 130
 XT keyword
 SORT command 146
 XWDCMP process option
 SuperCE utility (option 3.13) 382

Y

YES parameter
 ACTIVE field 316, 318, 320
 BROWSE Output field 186
 CONFIRM DELETE REQUEST field 127
 IF PARTITIONED, REPLACE LIKE-NAMED
 MEMBERS field 117
 LOCK field 36
 MIXED MODE field 51
 MULTIPLE STRINGS field 200
 NEWAPPL field 289
 NEWPOOL field 289

PACK DATA field 122
 PASSLIB field 290
 PURGE DATA SET field 114.2
 SPECIFY PACK OPTION FOR "TO" DATA SET
 field 117
 TABLE AVAILABLE field 307
 TABLE ON DISK field 309
 YYYY.DDD or YY.DDD format
 EXPIRATION DATE field 113
 YYYY/MM/DD or YY/MM/DD format
 EXPIRATION DATE field 113

Z

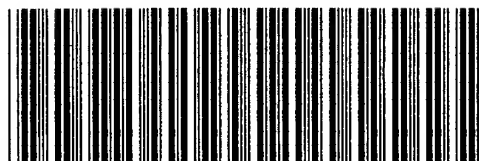
Z (compress data set) line command
 Data Set List utility (option 3.4) 141
 ZCMD input field 164
 ZDLBLKSZ variable 134
 ZDLCAT variable 135
 ZDLCDATE variable 134
 ZDLCONF variable 134
 ZDLDEV variable 134
 ZDLDSN variable 134
 ZDLDSORG variable 134
 ZDLEDATE variable 134
 ZDLEXT variable 134
 ZDLCMD variable 134

ZDLLRECL variable 134
 ZDLMSG variable 135
 ZDLNDSN variable 135
 ZDLRDATE variable 134
 ZDLRECFM variable 134
 ZDLREF variable 135
 ZDLsize variable 134
 ZDLUSED variable 134
 ZDLVOL variable 134

Numerics

1 to 9999 parameter
 SCROLL field 8
 3277KN parameter
 TERMINAL TYPE field 65
 3278CF parameter
 TERMINAL TYPE field 65
 3278KN parameter
 TERMINAL TYPE field 65
 3850 virtual volume, allocating a data set to 111
 3850 virtual volumes, accessing 32
 5550 terminal 166
 60 lines per page
 Lines per page field 67
 80 lines per page
 Lines per page field 67

SN28-4654-00





COMPAREX[®]

The intelligent comparison utility



**STERLING
SOFTWARE**

Systems Software Marketing Division

COMPAREX contains extensive keywords and options that move it into the realm of intelligent tools. Where most comparison utilities allow simple comparison of source code or sequential files, COMPAREX compares the contents of virtually **any** two files of similar or differing formats.

COMPAREX

offers comprehensive support for:

- System development
- Maintenance programming
- Conversions
- Vendor release updates
- Reconciliation of production libraries
- Test file generation
- Trouble-shooting and analysis

1

2

3

4

Provides Editor
Debugger and Unit Testing

Compares Two Files of
Any Format or Organization
and Isolates Differences

5

6

7

8

Maximizes Use of System
and Personnel Resources

Runs Extremely Fast
and is Easily Designed
For Ease of Use

COMPAREX's "intelligent" comparison features include:

- Multiple key specifications (synchronization)
- Field selects and masks
- Record filter (in and out) options
- Character squeezes (elimination of spaces, etc.)
- Desensitization of packed, binary and zoned fields
- Differentiation between data (keyed) files and text (unkeyed) files
- Conversion from one format to another (i.e. ISAM to VSAM)

Highlights All Changes

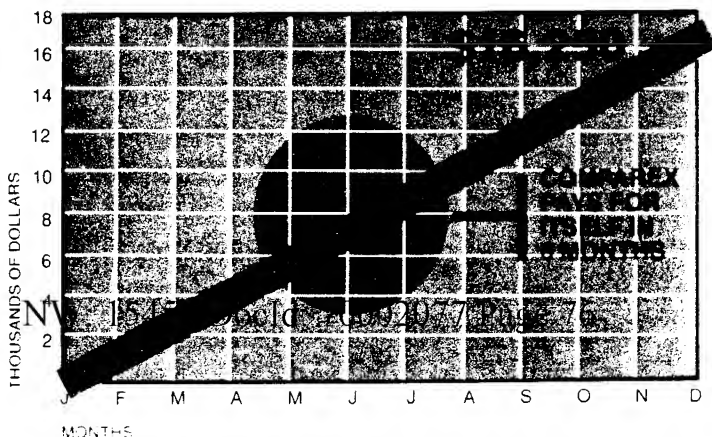
COMPAREX prints in either alphanumeric or hexadecimal display — and highlights **all** changes made. With this unique highlighting feature, checking is only a matter of flipping through the listing and seeing the changes clearly marked in front of you.

Supports All Major File Organizations

COMPAREX supports all major File organizations including VSAM, ISAM, sequential, partitioned data sets, and data base management files, including IMS, IDMS, ADABAS, RAMIS II, and more. COMPAREX allows comparison of JCL, source code, object code, and load modules. There are no restrictions for record length, file size, or block size, except those imposed by IBM architecture.

SAVINGS USING COMPAREX

COMPAREX gives you substantial savings in programmer time and expense. For example, suppose you have 25 programmers, each earning \$15.00 per hour. COMPAREX will save them at least 10 minutes a day in desk checking time alone. This translates into a savings of \$1,354 for one month... and after a year, that adds up to a savings of \$16,250!



Easy to Use

You can run COMPAREX in batch and in foreground using the ISPF interface with help screens. ISPF will lead you through the development of customizing parameters (keywords) when comparing two files. When you need it, a tutorial facility may be invoked by pressing a single key. You can review the output on the screen or queue it to your printer.

COMPAREX's "intelligent" comparison features include:

- **Multiple key specifications (synchronization)**
- **Field selects and masks**
- **Record filter (in and out) options**
- **Character squeezes (elimination of spaces, etc.)**
- **Desensitization of packed, binary and zoned fields**
- **Differentiation between data (keyed) files and text (unkeyed) files**
- **Conversion from one format to another (i.e. ISAM to VSAM)**

Isolates and Reconciles Changes

Instead of pouring through reams of listings to identify data and text file changes — and their results — you can use COMPAREX to isolate changes. Then you can easily reconcile them to the changes requested.

Creates Audit Trail

When source listings, JCL, and regression test results are submitted for review and approval, the changes are easily documented via COMPAREX reports.

Highlights All Changes

COMPAREX prints in either alphanumeric or hexadecimal display — and highlights **all** changes made. With this unique highlighting feature, checking is only a matter of flipping through the listing and seeing the changes clearly marked in front of you.





Supports All Major File Organizations

COMPAREX supports all major File organizations including VSAM, ISAM, sequential, partitioned data sets, and data base management files, including IMS, IDMS, ADABAS, RAMIS II, and more. COMPAREX allows comparison of JCL, source code, object code, and load modules. There are no restrictions for record length, file size, or block size, except those imposed by IBM architecture.

Easy to Use

You can run COMPAREX in batch and in foreground using the ISPF interface with help screens. ISPF will lead you through the development of customizing parameters (keywords) when comparing two files. When you need it, a tutorial facility may be invoked by pressing a single key. You can review the output on the screen or queue it to your printer.

USE COMPAREX TO:

-  Compare new release of software to a prior release for debugging and documentation purposes
 -  Compare source code versions updated through ISPF to provide a necessary audit trail
 -  Compare a data base against backup files to assure the integrity of the data
 -  Compare data output files to ensure that the results of source changes are as requested
-

Comparison Time Reduced 50% to 80%

COMPAREX users report typical "debugging" (desk checking) time is reduced by 50 percent... Others estimate savings as high as 80 percent. Increased productivity results because programmers have more time for pure program development.

Operating Environment

COMPAREX runs — without modification — in all IBM mainframe operating environments, such as DOS/VSE, VM/CMS, and OS (including MVS-XA).

COMPAREX needs only a minimum 100 K of virtual memory. At execution time, the block size of the files involved (and the optional invocation of the COMPAREX interface for PANVALET or LIBRARIAN files) dictate how large the partition size should be — usually about 150K.

No special data sets are required in order to execute COMPAREX.

Easy Installation — Requires No Modifications or Hooks

COMPAREX is easy to install. It requires no modifications to the operating system and constitutes a single load module in a single library.

Text Comparison Reports

Text Files

A text file is any data set with no inter-relationship of records through keys. Text files include source code, JCL, and documentation. In text mode, COMPAREX:

- Shows the differences by enclosing them in a frame
- Resynchronizes after inserts and deletes

Figure 1 illustrates a Difference Report resulting from a comparison of an original COBOL source program to an updated version.

There are options included with COMPAREX. In figure 1, the options that have been used are:

- The sequence numbers in columns 1 through 6 and 73 through 80 are masked during the comparison. Sequence numbers do not play a role in synchronizing the files, but they are displayed for reporting information.
- Blanks are squeezed from the comparison.
- File 1 is printed in full, allowing the user to see the changes in context.

Figure 2 illustrates another reporting feature, in the default mode, when COMPAREX shows only the differences.

FIGURE 2: TEXT DIFFERENCES — ONLY REPORT

[illegible]

COMPAREX clearly shows replacements

COMPAREX clearly shows insertions

COMPAREX clearly
framed differences

Figure 3 shows a sample COMPAREX Differences Report that compares a fixed length VSAM file versus a fixed length QSAM (flat) file. Note that the report is in hexadecimal "DUMPS" format but that vertical hex "DITTO" format or character format is also available. Also note the underscoring of differences and isolation of key mismatches (inserted or deleted records). Many free-form keywords are available to customize the comparison and vary the report format.



Effective, Reliable Customer Support

Sterling Software's reputation is built on customer service and technical support. In order to maintain a cohesive and informed user base, we provide:

- Clear technical documentation for on-site instruction
- Technical support problem-solvers who are easy to reach — just a Toll Free phone call away
- On-going follow-up by product representatives who identify and assist with special applications
- An active user solicitation plan to evaluate user-requested enhancements

30 Day Evaluation

Evaluate COMPAREX at your site with our free, 30-day trial program. This allows you to apply COMPAREX to your own requirements and evaluate its performance. There is no obligation, and once you've tried COMPAREX, you'll wonder how you ever did without it.

For a FREE, on-site evaluation of COMPAREX, at no obligation, call TOLL FREE, 800-824-8512. In California, Hawaii, Alaska and Canada call collect 916-635-5535.

800-824-8512



NW Systems Software Marketing Division
11050 White Rock Road #100
Rancho Cordova, CA 95670-6095 • 916/635-5535

COMPAREX is a registered trademark of Serena Consulting.

COMPAREX[®] 6.1.0 (87/054)

Installation

Guide

OS

Marketing Representative:

STERLING SOFTWARE
Systems Software Marketing Division
11050 White Rock Road
Rancho Cordova, California 95670-6095
Phone: (916) 635-5535
Telex: 377314

Copyright © SERENA Consulting 1987.
Licensed Material.

COMPAREX[®] is a Registered Trademark of:

SERENA Consulting
Post Office Box 117039
Burlingame, California 94010
Phone: (415) 347-0100
Telex: 172040 HQ SMT

Table of Contents

Background	1
Mainframe	1
Memory	1
Auxiliary Storage	1
Distribution Tape Contents	1
Installation Procedures	3
Find a Library	3
Prepare the Tape	3
Nonstandard Processing	4
Run the Job Stream	5
Installation Tailoring - Optional	5
COMPAREX Interface (CPXIFACE)	6
PANVALET and LIBRARIAN Release Levels	6
Edit CPXIFACE	6
ROSCOE Consideration	8
IDMS Exit	8
Multiple Interfaces	8
Generating Another Interface	8
Invoking Another Interface	8
External Modules	8
Assemble and Link Edit CPXIFACE	9
Installation of COMPAREX/ISPF	9
ISPF Release Level	9
Unload ISPF Datasets From Tape	9
Customize the SYSGEN Library	10
Linkage Edit CPX\$ISPF	11
Allocate a TABLE Library	12
Prepare for SYSGEN	12
For IBM Users Only	13
Sysgen the ISPF Interface	13
Connect To Your Main Menu	14
For FUJITSU/FACOM Users Only	15
Sysgen the ISPF Interface	15
Connect To Your Main Menu	16
Post Installation Procedures	17
Execute COMPAREX in the Background	17
Execute COMPAREX in the Foreground	19
Appendix A. Csect Source Code - Installation Tailoring	21
Installation Defaults	21
EBCDIC Translate Table	23
Appendix B. CPXIFACE	24
Other Interfaces	24
Assembly	25
Appendix C. ISPF Related Members	27
Link CPX\$ISPF	28
IBM Model Clist - Allocation for Sysgen	29
IBM Model Clist - Normal Allocation	30

List of Illustrations

Figure 1. Linkage Edit and Unload IFACE PDS	4
Figure 2. Object Code From Disk	4
Figure 3. Nonstandard Tape Override	5
Figure 4. Nonstandard Tape Override	5
Figure 5. ISPF Panel VENDOR in somnode.COMPAREX.PANELS	14
Figure 6. End of Job Messages	17
Figure 7. POSTINST in somnode.COMPAREX.IFACE	18
Figure 8. CPX@PRIM in somnode.COMPAREX.PANELS	19
Figure 9. CPXOPDS1 in somnode.COMPAREX.PANELS	20
Figure 10. Panel Invocation of COMPAREX Interface	20
Figure 11. COMPAREB in somnode.COMPAREX.IFACE	22
Figure 12. Source Code for CSECT COMPAREE	23
Figure 13. List of OTHer Interfaces	24
Figure 14. CPXIFASM in somnode.COMPAREX.IFACE	26
Figure 15. CPXISPF in somnode.COMPAREX.IFACE	28
Figure 16. LNKISPF in somnode.COMPAREX.IFACE	28
Figure 17. IBMAL1 in somnode.COMPAREX.CLIST	29
Figure 18. IBMAL2 in somnode.COMPAREX.CLIST	30

Background

This guide describes the systems requirements, the format of the distribution tape, the procedures for installation, and the procedures for post-installation verification. The guide assumes the reader has a working knowledge of his or her operating environment.

If you have need for further assistance in the installation of COMPAREX, contact your marketing representative as noted on the initial page of this document.

Mainframe

COMPAREX runs, without modification, on all IBM and plug compatible mainframes running all derivatives of OS/VS including MFT, MVT, VS1, MVS, MVS/XA and FACOM's OSIV/F4.

Memory

COMPAREX requires a minimum of 96K of virtual memory. At execution time, the block sizes of the files involved and the number of buffers (as specified by the BUFNO parameter on the //DD statement) influence this minimum size. The optional invocation of the COMPAREX interface (CPXIFACE) for PANVALET, LIBRARIAN and OTHER files further dictate how large the region should be.

COMPAREX is written in reentrant assembler and as such may be given the 'RENT' attribute at linkage edit time. It is eligible for inclusion in the Link Pack Area (LPA) so that concurrent users may share the same copy in storage. It is not eligible for inclusion above the 16 Meg (AMODE=24,RMODE=24) line of MVS/XA vet.

Auxiliary Storage

COMPAREX is a single load module in a single library. The load module size is about 60K. This can be held on three tracks of a 3380 disk pack. No special data sets are required by COMPAREX for its execution unless the optional ISPF interface is used. This can be held on less than two cylinders of a 3380 disk pack.

The optional COMPAREX interface (CPXIFACE) is another load module. The size of CPXIFACE is dependent on how it is generated but it seldom is larger than 60K.

Distribution Tape Contents

The distribution tape is a standard labeled 9-track tape. It contains the following datasets:

1. COMPAREX.OBJECT - object code to be link-edited into your Load Library - FB/80/3200
2. CPXIFACE.IFACE - CPXIFACE: source code and JCL to assemble and link; ISPF Interface JCL; IDMS Exit module
3. COMPAREX.PANELS - Panel library
4. COMPAREX.MSGS - Message library
5. COMPAREX.CLIST - Clist library
6. COMPAREX.SYSGEN - SYSGEN library
7. COMPAREX.CPX\$ISPF - Object code library for module CPX\$ISPF

The tape density is usually 6250 BPI. Unless 1600 BPI has been explicitly requested, it is cut at the 6250 density. If the density recorded on the distribution tape cannot be processed at the customer's site, call your marketing representative immediately.

Note: The User Reference Manual is no longer put on the distribution tape for fully paid licensees. All documentation has been converted to LaserScript¹ and is applicable only to AT class personal computers.

¹ Desktop Publishing Software from Command Technology Corp.

Installation Procedures

These are the steps to follow to install COMPAREX at your installation.

Find a Library

Determine which library you want COMPAREX to reside in. The library must be a partitioned data set (PDS). The record format should be undefined (RECFM=U).

Prepare the Tape

Pull any write ring from the distribution tape. Do not risk writing over the information on the distribution tape. Check the tape into your tape library. Note here the volume serial number (VOL=SER) written on the peel-off external label:

VOL=SER= _____

The following steps assume that you will mount the tape twice to unload it. It is conceivable that you could do the entire operation with a single tape mount. To accomplish this, you must combine the steps in "Linkage Edit and Unload IFACE PDS" on page 4 with "CPXISPF in somnode.COMPAREX.IFACE" on page 28 into a single job (*with appropriate usage of RETAIN*) and submit. This assumes, of course, that all of this JCL must be created by hand.

Prepare a job based on "Linkage Edit and Unload IFACE PDS" on page 4.

Change line 01 to use your standard //jobname JOB statement.

Change line 06 to point to the correct SYSDA, disk device used for work space during the linkage edit.

Change line 09 to show the selected library name.

Change line 11 and 19 to point to the correct type of tape drive with the // UNIT = statement. This tape drive is used for reading the distribution tape.

Change lines 12 and 20 to show the volume serial number.

JCL Statement	Line Number
//jobname JOB accounting info.class, etc.	(01)
//LKED EXEC PGM=LINKEDIT,REGION=512K,	(02)
// PARM='LIST,XREF,LET,RENT,SIZE=(256K,128K),NCAL'	(03)
//SYSPRINT DD SYSOUT=*	(04)
//SYSLIB DD DUMMY,DCB=BLKSIZE=80	(05)
//SYSUT1 DD UNIT=sysda, shop dependent	(06)
// SPACE=(CYL,1)	(07)
//SYSIMOD DD DISP=SHR,	(08)
// DSN=somnode.COMPAREX.LOADLIB	(09)
//SYSLIN DD DISP=OLD,DSN=COMPAREX.OBJECT,	(10)
// UNIT=tape, shop dependent for 1600/6250 BPI	(11)
// VOL=(,RETAIN,SER=_____),LABEL=(1,SL)	(12)
// DD DDNAME=SYSIN	(13)
//SYSIN DD *	(14)
NAME COMPAREX(R)	(15)
//IEBCOPY EXEC PGM=IEBCOPY,REGION=256K	(16)
//SYSPRINT DD SYSOUT=*	(17)
//SYSUT1 DD DISP=OLD,DSN=COMPAREX.IFACE,	(18)
// UNIT=tape, shop dependent for 1600/6250 BPI	(19)
// VOL=SER=_____,LABEL=(2,SL)	(20)
//SYSUT2 DD DISP=(,CATLG),DSN=somnode.COMPAREX.IFACE,	(21)
// UNIT=sysda,SPACE=(TRK,(30,10,20))	(22)
//SYSIN DD DUMMY	(23)

Figure 1. Linkage Edit and Unload IFACE PDS

Nonstandard Processing

If your installation standard requires that all object code be modified, such as with:

SETSSI yyddnnnn

Copy the object code from the distribution tape to a disk data set with the IBM utility IEBGENER and change lines 10 through 12 to point to the disk copy of the object code with a disposition of MOD, such as:

//SYSLIN DD DISP=MOD,DSN=DISK.COMPAREX.OBJECT	(10)
// UNIT=sysda, shop dependent	(11)
// VOL=SER=_____	(12)

Figure 2. Object Code From Disk

Use the VOL=SER of the disk pack instead of the volume serial number written on the distribution tape.

If your installation cannot process foreign tapes (standard labeled tapes created outside the installation), you will need to invoke bypass label processing by replacing lines 10 through 12 with:

```
//SYSLIN DD DISP=OLD,DSN=COMPAREX.OBJECT, (10)
// UNIT=tape, shop dependent for 1600/6250 BPI (11)
// LABEL=(2,BLP), (11A)
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200), (11B)
// VOL=SER=_____ (12)
```

Figure 3. Nonstandard Tape Override

The VOL=SER=_____ on line 12 is not the volume serial number of the tape but it is the **ALIAS** volume serial number assigned by your tape librarian.

Similarly, invoke bypass label processing by replacing lines 18 through 20 with:

```
//SYSUT1 DD DISP=OLD,DSN=COMPAREX.IFACE, (18)
// UNIT=tape, (19)
// LABEL=(5,BLP), (19A)
// DCB=(RECFM=VS,BLKSIZE=6020), (19B)
// VOL=SER=_____ (20)
```

Figure 4. Nonstandard Tape Override

Run the Job Stream

Run the job stream as modified in preceding steps. Examine the linkage editor printout. There should be no unresolved external reference and the listing should say that COMPAREX was placed in the library. If there is any unresolved external reference or if the listing does not say that COMPAREX was placed in the library, call your marketing representative immediately.

Installation Tailoring - Optional

COMPAREX, the load module, is composed of eight (*control sections*) CSECT's. Two of these may be tailored at the installation to modify the defaults.

Note: This is entirely optional and usually unnecessary.

Within library somnode.COMPAREX.IFACE, are two members:

- COMPAREB on page 22
- COMPAREE on page 23

that supply sample JCL and Assembler code to recompile these CSECT's and replace their counterparts permanently.

COMPAREX Interface (CPXIFACE)

It is recommended that the user perform the following steps. However, if your shop does not use PANVALET nor LIBRARIAN and does NOT expect to generate an interface to any other library/data management system, skip to the next section "Installation of COMPAREX/ISPF" on page 9.

PANVALET and LIBRARIAN Release Levels

PANVALET must be at release² 11.

LIBRARIAN must be at release level 3.2 or higher (3.1 will not work).

Edit CPXIFACE

Using TSO/ISPF, edit dataset somnode.COMPAREX.IFACE and select member CPXIFACE.

Approximately between the 60th and 110th statements are some global variables. Set the global variables in the assembler source code to:

```
*=====
* BELOW IS WHERE THE USER CAN TAILOR THE SOURCE FOR THE SHOP. *
*=====
&OPSYS      SETB  1          '&OPSYS' at OS and CMS
*OPSYS      SETB  0          '&OPSYS' at DOS
*CMS        SETB  1          '&CMS'   at CMS
&VERSION    SETC  '6'
&CPXREL     SETC  '1'
&CPXMOD     SETC  '0'
&CPXVRM     SETC  '&VERSION..&CPXREL..&CPXMOD'
&RELDATE    SETC  '87/054'
```

If your shop uses Pansophic's PANVALET, set:

```
&PAN        SETB  1          (YES)
```

else

```
&PAN        SETB  0          (NO)
```

² PANVALET release 12 through PACB is not supported yet

If your shop uses Applied Data Research's LIBRARIAN, set:

```
&LIB      SETB   1      (YES)
&LIB$CMC  SETC   'xxxx'  Your NUMERIC Current Management Code
&GEM      SETB   0      (NO)
```

else

```
&LIB      SETB   0      (NO)
```

The Current Management Code *LIB\$CMC* is optional. It is only needed by LIBRARIAN to access 'PROD2' modules. If you do not have PROD2 members or do not want users to access them, set the 'xxxx' above to '0000'. If the user has multiple LIBRARIAN Masters, each of which has its own 'Current Management Code', again set the 'xxxx' to zero. Dynamically, pass in the four-digit code via the PARM at execution time like this:

```
SYSUT1=(LIB,M=abc,PARM=1234)
```

Note: LIBRARIAN and GEM are mutually exclusive.

If your shop uses FUJITSU/FACOM's GEM, which is similar to LIBRARIAN, set:

```
&LIB      SETB   1      (YES/GEM)
&GEM      SETB   1      (YES)
```

else

```
&LIB      SETB   0      (NO)
```

If your shop has at least one of the file structures in "List of OTHER Interfaces" on page 24 and desires COMPAREX to process it directly, then set:

```
&OTH      SETB   1      (YES)
```

else

```
&OTH      SETB   0      (NO)
```

If you SETB &OTH to 1, then you may SETB **one and only one** of the global variables in "List of OTHER Interfaces" on page 24 to 1 also. Note in the comments (*beyond column 16*) area that only those marked

(NO)

or

(BETA)

may be chosen for use. If you choose one marked BETA, it will work in the environment where it was developed but has never been tested in a stressed environment. Generally, the BETA designation only lasts for a single *interim release* and then is changed to NO which implies a guarantee. Note that the BETA designation only affects the individual interface and is *insulated* from all other interfaces. If you have any qualms about trying new software, then stay away from the BETA interfaces. Those marked

(FUTURE)

are in varying stages of development and should not be used.

ROSCOE Consideration

ROSCOE users should do some extra editing. Find `ROSS2MAR` in column one. This is the logic entitled *Member Access Rights* which allows or disallows certain user ID's access to certain members on the library. As delivered, it is fairly restrictive. It does allow access to members with the *shared* attribute, but anything else is basically restricted. This logic is your responsibility to maintain.

IDMS Exit

IDMS users have a special COBOL exit module called `JOSEFINE`³ which may be used as a model for navigating through data bases.

If it is used, the compile step must incorporate the following attributes:

```
ENDJOB
NODYNAM
NORES
```

Multiple Interfaces

Some shops will want to have multiple interfaces available for processing at all times. While it is true that only one of the "OTH" interfaces can be generated at any one time, it doesn't mean that you cannot have separate modules (versions of `CPXIFACE`) for each function.

To accomplish this, you must compile and link a unique module name for each of them. We recommend that the unique name you choose start with the letters `CPX` so that it can be identified easily as being associated with `COMPAREX`.

Generating Another Interface

The only thing that needs to be changed is the linkage edit control card image in member `CPXIFASM`:

```
NAME    CPXIFACE (R)
```

Change it to another name such as:

```
NAME    CPXIDMS (R)
```

Invoking Another Interface

Now, when you want that interface, you must instruct `COMPAREX` to load and use "`CPXabcde`" as opposed to the default "`CPXIFACE`" by specifying:

```
CPXIFACE = CPXabcde
```

If you also want `DL/1`, generate "`CPXDLI`" and so on.

External Modules

Some of the `OTHER` interfaces need external modules, and some don't. At least one of the interfaces - `RAMIS` - needs a module (`RPIINT`) that is considered an extra cost item from the vendor. Following is a list of interfaces and their required module names.

- `ADABAS` needs module `ADAUSER`
- `CAMLIB` needs modules `USERIOCS`, `CAMPIOCS`, and `CAMLIOCS`
- `DL/1` needs module `CBLTDLI`

- IAM needs module LAMACC
- IDMS needs module IDMS
- OWL needs module OWLIEAV5
- RAMIS needs module RPIINT (RAMIS Procedural Interface)
- ROSCOE needs macro's for the Assembly, and system modules for the linkage edit

Assemble and Link Edit CPXIFACE

Using TSO/ISPF, edit member "CPXIFASM in somnode.COMPAREX.IFACE" on page 26. Turn CAPS ON first. This JCL assembles and linkage edits CPXIFACE. You may, however, prefer to use one of your standard procedures such as "ASMFCL" or "ASMHCL" instead.

In the LINKAGE EDIT step, the SYSLIB statement should point to the library containing:

PANVALET Access Methods (PAM)

and/or

LIBRARIAN File Access Interface Routines (FAIR)

and/or

The system routines for the OTHer interface

The SYSLMOD statement must point to the library where the COMPAREX load module resides:

'somnode.COMPAREX.LOADLIB'

Submit the job for execution. There should not be any assembly or linkage editing (*Weak EXTRNs excepted*) errors. Once COMPAREX 6.1.0 and CPXIFACE 6.1.0 have been generated onto the same partitioned library, testing may begin.

Installation of COMPAREX/ISPF

If your shop does not have TSO/ISPF or does not wish to use this ISPF interface, skip to "Post Installation Procedures" on page 17.

ISPF Release Level

Only the newer release levels will work with this interface. The older releases (such as 1.1.x) cannot handle the following commands:

```
)BODY width(&zscreenw) expand(!!)
%!-! COMPAREX/ISPF &CPXVRM Primary Menu !-!
```

Commands similar to the above may also be seen in Panel "ISPF Panel VENDOR in somnode.COMPAREX.PANELS" on page 14. ISPF 2.1 or better is required, otherwise, it is doubtful that this interface can be used.

This interface is panel driven. Prior releases⁴ of COMPAREX were program driven. Panel jumping is now allowed and even encouraged.

Unload ISPF Datasets From Tape

Using TSO/ISPF, edit member "CPXISPF in somnode.COMPAREX.IFACE" on page 28.

- Change the JOBCARD to installation values

⁴ Releases 5.0, 6.0, 6.0.1, and 6.0.2

- Change the parameter values on the 'PROC' statement
 - TAPEVOL - enter the supplied COMPAREX tape volume serial number.
 - DISKVOL - enter the volume name of the disk you want the datasets to reside on.
 - UNIT - enter a unit type, i.e., SYSDA, 3350, 3380, etc.
 - SOMNODE - enter a high-level index name of your choice.

Submit the job to execute.

Customize the SYSGEN Library

Edit the SYSGEN Library members (DSN=somnode.COMPAREX.SYSGEN).

- BACKGRND

The default value of this member is:

TSO-SUBMIT

If this is used as the input to the SYSGEN step of installing the ISPF Interface, then background jobs generated through the interface will be SUBMIT'ed (via the TSO command) for execution into the batch environment. Internally, module CPXSISPF dynamically allocates dataset:

&prefix.CPXydddd.Thhmmssst.CNTL

at his initialization time to hold each background job. When the job has been constructed (*and optionally presented for editing*) the TSO SUBMIT command is invoked using this dataset as its only input. The dataset is dynamically deleted after submission.

User exits to this TSO Command can plug in the User, Group and Password. An additional feature is that a status line detailing the submitted job *penetrates* the current panel:

JOB TS123DTA(JOB02497) SUBMITTED

An alternative to the TSO SUBMIT concept, is the Internal Reader. Modify the member to read:

INTERNAL-READER

If this is used as the input to the SYSGEN step of installing the ISPF Interface, then background jobs generated through the interface will be done by dynamically opening up an internal reader and writing the job to a JESn reader queue. It is very fast and efficient but creates administrative problems at a number of shops. For example, if one of the datasets to be compared is RACF protected, then the job card(s) must contain the User, Group, and Password. These can be put on the job card when presented in a panel, but it is a breach of security to expose the information to others looking over your shoulder or held in a non-RACF protected Profile Pool dataset.

- CPXDSN

Replace the default value:

somnode.COMPAREX.LOADLIB

with the name of the load library where COMPAREX resides.

It must be modified in all situations. If COMPAREX is installed in a private library, that library name must be reflected here. If COMPAREX is installed in a LINKLIST Library, then a //STEPLIB is not needed in a background job and the LINK is different in the foreground. Remember, COMPAREX is written in reentrant Assembler and is eligible for inclusion in the Link Pack Area.

To activate this alternative to referencing a private library, modify the member to read:

NULLFILE

or

DUMMY

or

< == = spaces - 44 consecutive blanks

- **DEFJOB**

Replace the default values with a JOB statement valid for your installation. Each of the four lines must start with //.

Note: The content of **DEFJOB** is presented to the user upon submission of the first COMPAREX batch job. The user can then modify the default JOB statement and it will be saved in the users profile.

- **DSPRINT**

The default value of this member is:

DSPRINT

If this is used as the input to the SYSGEN step of installing the ISPF Interface, then when a difference report, as generated from the foreground execution (Option 6), is to be printed:

REPORT DISPOSITION == => P

the TSO Command

DSPRINT

is used to print that report.

An alternative to using **DSPRINT** is the **PRINTOFF** Command. To activate this alternative to **DSPRINT**, edit the member to read:

PRINTOFF

or anything else you wish that uses the same syntax as the "PRINTOFF" command. Whatever name you choose, that will be what is executed when the foreground created difference report is to be printed.

Linkage Edit CPX\$ISPF

The object code for module CPX\$ISPF is supplied in library

somnode.COMPAREX.CPX\$ISPF

It must now be linkage edited into a load library (perhaps one already exists for this purpose). Using TSO/ISPF, edit member "LNKISPF in somnode.COMPAREX.IFACE" on page 28.

It is recommended that you **NOT** bring module ISPLINK together with CPX\$ISPF. It will result in an unresolved external reference, but that is OK. Internally, CPX\$ISPF will detect the absence of a resolved (*WEAK*) external reference and dynamically load the current ISPLINK from where DDNAME ISPLLIB points to. The advantage is that when release levels of ISPF change, no relinking of CPX\$ISPF is forced. The latest module will always be brought in dynamically.

If you choose to load ISPLINK dynamically as recommended, and your users generally do not allocate DDNAME ISPLLIB to their ISPF session, they will get a warning message immediately upon calling CPX\$ISPF:

ISPLLIB DD STATEMENT MISSING

This is not severe. It is only a nuisance. There are two ways to eliminate this. One is to relink module CPX\$ISPF and this time include ISPLINK. The other is to modify the source code⁵ to CPX\$ISPF such that global variable:

```
&NOLLIB SETB 1
```

is turned on and then recompile and link CPX\$ISPF. This global variable (&NOLLIB) stops CPX\$ISPF from opening ddname ISPLLIB to check for module ISPLINK. It automatically brings in ISPLINK from LINKLIST.

If you choose to statically linkage edit ISPLINK in with CPX\$ISPF, you will need to modify the job slightly. You will need to point SYSLIB to the library where ISPLINK resides e.g.

```
//SYSLIB DD DISP=SHR,
// DSN=SYS1.ISPLLIB <=== Where ISPLINK is
```

and since the external reference to ISPLINK is weak, you must specifically include the module e.g.

```
//SYSIN DD *
        ENTRY CPX$ISPF
        INCLUDE SYSLIB(ISPLINK)
        NAME CPX$ISPF(R)
```

FUJITSU/FACOM users generally do not have module 'ISPLINK' available to them. They do have module 'PFDLINK' which serves the same purpose. It is recommended that these users assign an alias to PFDLINK called ISPLINK, or copy PFDLINK to another library and rename that to ISPLINK.

Allocate a TABLE Library

Using option 3.2 of TSO/ISPF, allocate a TABLE library. The name of the library should conform to the other installed COMPAREX libraries (*somnode.COMPAREX.TABLES*). The DCB for this library is DSORG=PO and LRECL=80. Use the minimum size (one track) for this library.

Prepare for SYSGEN

Using TSO/ISPF, edit the CLIST library (*somnode.COMPAREX.CLIST*). Note that there are six members there:

1. **COMPAREX** - Invokes COMPAREX in the foreground
2. **FACOMAL1** - Allocates libraries for SYSGEN (FACOM only)
3. **FACOMAL2** - Allocates libraries for ISPF Interface (FACOM only)
4. **IBMAL1** - Allocates libraries for SYSGEN (IBM only)
5. **IBMAL2** - Allocates libraries for ISPF Interface (IBM only)
6. **SYSGEN** - Create member CPXINOPT in Table library

In order to save profiles and load previously created profiles, the PROFILE datasets MAY need to be altered. The PROFILE dataset should be copied via IEBCOPY to another and reblocked to at least 6000 bytes per block.

If you are using an IBM operating system such as VS1, MVS, or MVS/XA; follow the directions "For IBM Users Only" on page 13 and then skip to the "Post Installation Procedures" on page 17.

If you are using OSIV/F4 from FUJITSU/FACOM; follow the directions "For FUJITSU/FACOM Users Only" on page 15 and then skip to the "Post Installation Procedures" on page 17.

⁵ Paid licensees receiving a custom tape get the source code in 'somnode.COMPAREX.IFACE'

For IBM Users Only

Edit member "IBMAL1 in somnode.COMPAREX.CLIST" on page 29. Set CAPS ON first. Modify it to reflect your ISPF Panel library in concatenation with the newly created 'somnode.COMPAREX.PANELS'. Also reflect the accurate name of 'somnode.COMPAREX.TABLES'. Now save the member.

Exit from TSO/ISPF into native TSO (*get a READY*) and execute the CLIST IBMAL1:

```
EXEC 'somnode.COMPAREX.CLIST(IBMAL1)'
```

Note: If you use your own version of IBMAL1 (*not the supplied version*) you may get yourself in trouble if the concatenation under DDNAME ISPTLIB already contains member CPXINOPT.

Sysgen the ISPF Interface

Get back⁶ into ISPF option 6 from TSO and generate the ISPF interface:

```
EXEC 'somnode.COMPAREX.CLIST(SYSGEN)'
```

When prompted, enter the name of the SYSGEN library:

```
'somnode.COMPAREX.SYSGEN'
```

System generation will be performed and member CPXINOPT is added⁷ to the installation ISPF table library:

```
(DDNAME=ISPTABL,DSNAME=somnode.COMPAREX.TABLES).
```

If you receive any errors in performing the SYSGEN CLIST such as:

```
WRITE SERVICE FAILED - CPXDSN
```

then you must retrace your steps all the way back to "Allocate a TABLE Library" on page 12. The partially created TABLE library is crippled, must be deleted, and reallocated.

Edit member "IBMAL2 in somnode.COMPAREX.CLIST" on page 30. Set CAPS ON first. Modify it to reflect your ISPF Panel library in concatenation with the newly created 'somnode.COMPAREX.PANELS'. Similarly, take care of the MSGS, TABLES, and the LOAD library where module CPXSISPF resides. Now save the member.

It is recommended that the TABLES library be copied (*blended*) into an existing library that most users already have DDNAME ISPTLIB directed to. This will require the least amount of disruption to existing LOGON procedures. In a similar fashion, the PANELS, MSGS, and LOAD libraries could be merged into existing libraries.

The order of concatenation is important. Remember that the library with the larger block size must be first. If you have merged the supplied libraries with your own, this step (*allocation by concatenation*) is not necessary. If it is necessary to do this allocation by concatenation, exit TSO/ISPF one more time and execute this CLIST:

```
EXEC 'somnode.COMPAREX.CLIST(IBMAL2)'
```

⁶ Do not LOGOFF - just key in "ISPF"

⁷ Actually four tables are within this single member

Connect To Your Main Menu

The supplied panel library contains members **ISR@PRIM** and **VENDOR**. They suggest that the method of gaining entry to COMPAREX is to key in **V** on the main menu (*ISR@PRIM*) and go to panel **VENDOR** which already is set to invoke the COMPAREX/ISPF Interface. "ISPF Panel **VENDOR** in somnode.COMPAREX.PANELS" can be expanded on to include other software in the future.

Note: One of your selection menus can be modified instead to invoke COMPAREX.

Using TSO/SPF, edit the IBM supplied ISPF panel library member **ISR@PRIM** to add COMPAREX as a selection. The selection character invokes CPX\$ISPF in one of these ways:

```
' PANEL(CPX@PRIM) NEWAPPL(CPX) NEWPOOL '  
or  
' PANEL(CPX@PRIM) NEWAPPL NEWPOOL '  
or  
' PANEL(CPX@PRIM) '
```

Note that invocation is for a PANEL. Earlier releases invoked a PROGRAM (CPX\$ISPF). This may be confusing. Beware.

```
)BODY width(&zscreenw) expand(!!)  
%!-! Vendor Supplied Software !-!  
%Option ==>_ZCMD +  
%  
% 1 +COMPAREX - Invoke%COMPAREX/ISPF  
% 2 +CHANGE MAN - Invoke%Change Man+Services  
% X +EXIT -%Exit+to primary option menu  
%  
+Enter%END+command to return to primary option menu.  
%  
)PROC  
  &ZSEL = TRANS( TRUNC (&ZCMD, '.' )  
    1, 'PANEL(CPX@PRIM) NEWAPPL(CPX) NEWPOOL '  
    2, 'CMD(CMNINIT) NEWAPPL(CMN) NEWPOOL '  
    , ' , ' , '  
    X, 'EXIT'  
    *, '?' )  
)END
```

Figure 5. ISPF Panel **VENDOR** in somnode.COMPAREX.PANELS

The implication of using variations on NEWAPPL/NEWPOOL is that it insulates this application (and shared variable pool) from any other concurrent application (on another screen perhaps). It is recommended that

```
' PANEL(CPX@PRIM) NEWAPPL(CPX) NEWPOOL '
```

be used, but it does require a small increase in resource. For further information on this subject, refer to the IBM publication:

Note: NEWAPPL and NEWPOOL may affect your PFK settings, particularly if you use other than the IBM standard.

Now invoke the interface (through panel VENDOR or direct depending on how it was installed) to see if you can get to "CPX@PRIM in somnode.COMPAREX.PANELS" on page 19. Once into this initial panel, you may *jump* around to others in the normal ISPF fashion within the first three (0, 1, or 2) options. If you pick one of the others (3, 4, 5, 6, 7, or 8), you invoke program CPX\$ISPF.

If youabend immediately withabend code DB4, it is because CPX\$ISPF cannot open (TBOPEN) the table library which is pointed to by DDNAME ISPTLIB. Internally, module CPX\$ISPF makes many calls to module ISPLINK at initialization time. Each call (*defining the variables, VPUT to shared pool, etc*) is counted by adding decimal 100 to an internal counter. If the return code from any one call is non-zero, the sum of the calls (times 100) plus the return code (usually 8) is the decimal code for the intentionalabend. In this case, hexadecimal 'DB4' is decimal '3508' meaning that the thirty-fifth call received a return code of eight. One otherabend code that occurs infrequently is E7C or decimal '3708' which means that table member DSPRINT could not be found in the table library. This occurs when a SYSGEN was done using an older (from release 4.4 or 5.0) SYSGEN library when member 'DSPRINT' was not part of the process.

Once under the control of program CPX\$ISPF, panel jumping is prohibited. The impact of doing so is that dialogue variables become *stacked*⁸. Also, ending the current panel with the RETURN command (*or equated function key e.g. PFK 4*); takes you all the way out to the primary panel and leaves dialogue variables hanging.

The installation of COMPAREX/ISPF is now complete. Go to "Post Installation Procedures" on page 17.

For FUJITSU/FACOM Users Only

Edit member FACOMAL1. Modify it to reflect your PFD Panel library in concatenation with the newly created 'somnode.COMPAREX.PANELS'. Also reflect the accurate name of 'somnode.COMPAREX.TABLES'. Now save the member.

Exit from PFD into native TSS and execute the CLIST "FACOMAL1":

```
EXEC 'somnode.COMPAREX.CLIST(FACOMAL1)'
```

Sysgen the ISPF Interface

Get back into PFD option 6 from TSS and generate the ISPF interface:

```
EXEC 'somnode.COMPAREX.CLIST(SYSGEN)'
```

When prompted, enter the name of the SYSGEN library:

```
'somnode.COMPAREX.SYSGEN'
```

System generation will be performed and member CPXINOPT is added⁹ to the installation ISPF table library:

```
(DDNAME = PFDTABL,DSNAME = somnode.COMPAREX.TABLES).
```

If you receive any errors in performing the SYSGEN CLIST such as:

```
WRITE SERVICE FAILED - CPXDSN
```

then you must retrace your steps all the way back to "Allocate a TABLE Library" on page 12. The partially created TABLE library is crippled, must be deleted, and reallocated.

⁸ May cause unpredictable results in future panels

⁹ Actually four tables are within this single member

Using PFD, edit the CLIST library (somnode.COMPAREX.CLIST). Edit member FACOMAL2. Modify it to reflect your ISPF Panel library in concatenation with the newly created 'somnode.COMPAREX.PANELS'. Similarly, take care of the MSGS, TABLES, and the LOAD library where module CPX\$ISPF resides. Now save the member.

It is recommended that the "TABLES" library be copied (blended) into an existing library that most users already have DDNAME "PFDTLIBS" directed to. This will require the least amount of disruption to existing LOGON procedures. In a similar fashion, the PANELS, MSGS, and LOAD libraries could be merged into existing libraries.

The order of concatenation is important. Remember that the library with the larger block size must be first. If you have merged the supplied libraries with your own, this step (allocation by concatenation) is not necessary. If it is necessary to do this allocation by concatenation, exit PFD one more time and execute this CLIST:

```
EXEC 'somnode.COMPAREX.CLIST(FACOMAL2)'
```

Connect To Your Main Menu

Using PFD, edit the FACOM supplied panel library member JRR#POM to add COMPAREX as a selection. The selection character invokes CPX\$ISPF in one of these ways:

```
' PANEL(CPX@PRIM) NEWAPPL(CPX) NEWPOOL '  
or  
' PANEL(CPX@PRIM) NEWAPPL NEWPOOL '  
or  
' PANEL(CPX@PRIM) '
```

Note: NEWAPPL and NEWPOOL may affect your PFK settings, particularly if you use other than the FACOM standard.

The implication of using variations on NEWAPPL/NEWPOOL is that it insulates this application (and shared variable pool) from any other concurrent application (on another screen perhaps). It is recommended that

```
' PANEL(CPX@PRIM) NEWAPPL(CPX) NEWPOOL '
```

be used, but it does require a small increase in resource.

The supplied panel library contains members ISR@PRIM and VENDOR. They suggest that the method of gaining entry to COMPAREX is to key in 'V' on the main menu (ISR@PRIM) and go to panel 'VENDOR' which already is set to invoke the COMPAREX/ISPF Interface. VENDOR can be expanded on to include other software in the future. See "ISPF Panel VENDOR in somnode.COMPAREX.PANELS" on page 14.

Note: One of your selection menus can be modified instead to invoke COMPAREX.

The installation of COMPAREX/ISPF is now complete.

Post Installation Procedures

We will execute COMPAREX as a background submitted job stream and in the foreground through the ISPF Interface.

Execute COMPAREX in the Background

Edit member "POSTINST in somnode.COMPAREX.IFACE" on page 18. Submit the sample batch execution and review the printout. Near the end of the printout, find message CPX75I. It should read:

```
CPX75I - RECORDS PROCESSED: SYSUT1(4)/SYSUT2(4), DIFFERENCES(2,1,1)
        EXPLANATION - 2 RECORDS DIFFER THAT SYNCHRONIZED TOGETHER
                      1 RECORD WAS CONSIDERED INSERTED ON SYSUT1
                      1 RECORD WAS CONSIDERED INSERTED ON SYSUT2
```

Figure 6. End of Job Messages

If message CPX75I does not appear, call your marketing representative.

If message CPX75I appears but the numbers are different, examine the job stream to insure that the example job stream was correctly entered. If you are still unable to resolve the problem, call your marketing representative.

If the numbers in message CPX75I are the same as in the above message, you have successfully installed COMPAREX, and you have run the sample execution correctly. COMPAREX can now be used as documented in the COMPAREX reference manuals.

```

//jobname JOB (account),'POST INSTALL',          <=== Change Accordingly
//          CLASS=?,NOTIFY=?,                    <=== Change Accordingly
//          MSGCLASS=?                          <=== Change Accordingly
//*
//*****
//* This post-installation procedure will verify that COMPAREX      *
//* has been linkage-edited properly into your library.              *
//*****
//*
//COMPARE PROC
//COMPAREX EXEC PGM=COMPAREX,
//          REGION=128K
//STEPLIB DD DISP=SHR,
//          DSN=somnode.COMPAREX.LOADLIB <=== Change Accordingly
//SYSPRINT DD SYSOUT=*
//          PEND
//*
//SAMPLE EXEC COMPARE
//SYSUT1 DD *
KEY01
KEY02 Identical
KEY03
KEY10 DATA
//*
//SYSUT2 DD *
KEY01 Different
KEY02 Identical
KEY04
KEY10
//*
//SYSIN DD *
MAXDIFF=10,CONTINUE,HELP
KEY=(1,5) /* Use KEY Synchronization - remember, relative to ONE */
FORMAT=12 /*Since there is no hexadecimal data, see alpha only */
MASK=(73,END) /*Ignore any sequence number in the last 8 bytes */
CASE=mixed /*There are lower case characters, see them */
//*

```

Figure 7. POSTINST in somnode.COMPAREX.IFACE

Execute COMPAREX in the Foreground

Using TSO/ISPF, get from the main menu into the primary menu of COMPAREX. This depends on how the connection was made (VENDOR menu or a direct call of CPX@PRIM from the main menu). Refer to "CPX@PRIM in somnode.COMPAREX.PANELS".

Enter 6 at Option ==> to invoke COMPAREX in the foreground. You will be prompted to enter a dataset name for file SYSUT1 as in "CPXOPDS1 in somnode.COMPAREX.PANELS" on page 20.

Note: For this example, we recommend that you try to keep it simple and limit your comparison to two files that have very few, if any, differences.

```
..... COMPAREX/ISPF 6.1.0 Primary Menu .....
Option ==>

0 - SHORT CUT   - Single screen for options and dataset names
1 - OPTIONS     - Specify compare options for this session
2 - DS NAMES    - Specify dataset names to be compared
3 - SAVE       - Save options profile for future sessions
4 - LOAD       - Select/Delete options profile from prior session(s)
5 - CLEAR      - Clear previously loaded profile
6 - FOREGROUND - Invoke COMPAREX in the foreground and wait
7 - BACKGROUND - Invoke COMPAREX as a submitted batch job
8 - BACKGROUND - Similar to above but edit job (and optionally SUBMIT)
X - EXIT       - Exit

Press HELP KEY for tutorial assistance at any point;
Enter END Command to exit.
```

Figure 8. CPX@PRIM in somnode.COMPAREX.PANELS

Fill in the name of a known dataset in the normal ISPF fashion and hit enter. Assuming you did not make any errors in dsname syntax, you will be prompted by panel 'CPXOPDS2' (which is very similar to 'CPXOPDS1'), for the file name of SYSUT2. Fill in the name of another known dataset or even the same name as that of SYSUT1 and hit enter.

```

..... File One (SYSUT1) .....
Command ==>

ISPF LIBRARY:
PROJECT ==> _____
LIBRARY ==> _____
TYPE ==> _____
MEMBER ==> _____ DATASET PASSWORD ==> _____ (If Protected)

OTHER PARTITIONED OR SEQUENTIAL DATASET:
DATASET NAME ==> _____
VOLUME SERIAL ==> _____ (If Not Cataloged)
UNIT ==> _____ (If Not Cataloged)

TYPE OF DATASET ==> _____ (COMPAREX Interface Only)
1 - PANVALET 2 - LIBRARIAN/GEM 3 - OTHER PROPRIETARY LIBRARY/DBMS
MEMBER NAME ==> _____ (16 Character Member Name)
INCLUDES ==> _____ (YES or NO - Expand Includes)
LEVEL ==> _____ (LIBRARIAN Only - Archie Level)
PARAMETER ==> _____ ('Parameter Data' for CPXIFACE)

Press ENTER to register; Enter END Command to exit.

```

Figure 9. CPXOPDS1 in somnode.COMPAREX.PANELS

COMPAREX should now be running in the foreground comparing the two datasets you just specified. Since we did not specify any options to the compare, there may be many differences particularly if you specified datasets that are vastly different. When COMPAREX finishes executing, the difference report will be presented for browsing. At this point we are certain that the ISPF interface to COMPAREX is functional.

If you force invocation of CPXIFACE in the foreground by specifying PAN, LIB, or OTH for SYSUT1 or SYSUT2, e.g.

```

TYPE OF DATASET ==> 1 (COMPAREX Interface Only)
1 - PANVALET 2 - LIBRARIAN/GEM 3 - OTHER PROPRIETARY LIBRARY/DBMS

```

Figure 10. Panel Invocation of COMPAREX Interface

and you have not:

- allocated DDNAME ISPLLIB to where COMPAREX and CPXIFACE reside
- nor put COMPAREX and CPXIFACE in a LINKLIST library

then COMPAREX will not be able to load CPXIFACE and an S806 abend will occur.

Appendix A. Csect Source Code - Installation Tailoring

Installation Defaults

```
//jobname JOB (account),'DEFAULTS',          <=== Change Accordingly
//          CLASS=?,NOTIFY=?,                <=== Change Accordingly
//          MSGCLASS=?                        <=== Change Accordingly
//*****
//* This is a sample Assembly of CSECT 'COMPAREB' which sets the *
//* installation defaults for any execution of COMPAREX. You may *
//* prefer to use one of your standard procedures such as 'ASMFCL' *
//* or 'ASMHCL' instead. Just be sure that it assembles clean and *
//* link-edits without any UNRESOLVED EXTERNAL REFERENCES.      *
//* ----- *
//*****
//ASMLINK PROC ASM=IEV90,          <==== Name of your Assembler (IFOX00)
//          IEWL=IEWL,            <==== Name of your Link Editor
//          UNIT=SYSDA,           <==== Disk unit type
//          SOMNODE='SYS2.' <==== Highest level node
//ASM      EXEC PGM=&ASM,REGION=512K,
//          PARM='LIST,XREF,RENT,?????' <=== Change Accordingly
//****    IEV90 would probably use PARM='LIST,XREF,RENT,OBJECT'
//****    IFOX00 would probably use PARM='LIST,XREF,RENT,OBJ' and
//****          DDname //SYSGO instead of //SYSLIN
//****
//SYSLIB   DD DISP=SHR,DSN=SYS1.MACLIB
//SYSPRINT DD SYSOUT=*
//SYSPUNCH DD DUMMY
//SYSUT1   DD UNIT=&UNIT,SPACE=(CYL,(1,1))
//SYSLIN   DD DISP=(,PASS),DSN=&&OBJECT, <=== Could be //SYSGO
//          UNIT=&UNIT,SPACE=(TRK,(2,1)),
//          DCB=(RECFM=F,BLKSIZE=80)
//SYSIN    DD DUMMY
//*
//LKED     EXEC PGM=&IEWL,PARM='LIST,XREF,MAP,RENT'
//SYSPRINT DD SYSOUT=*
//SYSLIB   DD DISP=SHR,
//          DSN=&SOMNODE.COMPAREX.loadlib <=== Change Accordingly
//SYSLMOD  DD DISP=SHR,
//          DSN=&SOMNODE.COMPAREX.loadlib <=== Change Accordingly
//SYSUT1   DD UNIT=&UNIT,SPACE=(CYL,1)
//SYSLIN   DD DISP=(OLD,DELETE),DSN=&&OBJECT
//          DD DDNAME=SYSIN
//          PEND
```

[Figure continued on next page]

[Figure continuation]

```
//*
//ASMLINK EXEC ASMLINK
//ASM.SYSIN DD *
COMPAREB CSECT *
*=====
* EVERY DC (DEFINE CONSTANT) MUST BE FOR 80 BYTES, COMPAREX ASSUMES *
* THAT THE CSECT IS COMPOSED OF 80-BYTE LINES DELIMITED BY X'FFFF'. *
*=====
DC CL80'*****'
DC CL80'* * *   I N S T A L L A T I O N   D E F A U L T S   * * *'
DC CL80'* * *                               B Y D D T , F E B 2 3 , 1 9 8 7   * * *'
DC CL80'*****'
DC CL80'          M A X D I F F = 5 0 , C O N T I N U E , H A L T = C O N D          '
DC CL80'*****'
DC CL80'*   E N D O F I N S T A L L A T I O N D E F A U L T S   *'
DC CL80'*****'
DC X'FFFF'          <=====VERY IMPORTANT!!!
          END
//*
//LKED.SYSIN DD *
          INCLUDE SYSLMOD(COMPAREX)
          ENTRY COMPAREX
          NAME COMPAREX(R)
//* EOJ
```

Figure 11. COMPAREB in somnode.COMPAREX.IFACE

EBCDIC Translate Table

COMPAREE	CSECT	*	SEPARATE *C*S*E*C*T*
EBCDICTB	DC	256C'.'	DEFAULT CHARACTER IS A PERIOD
	ORG	EBCDICTB+C' '	UP TO BLANK
	DC	C' '	THE BLANK
	ORG	EBCDICTB+C'A'	UP TO A
	DC	09AL1(*-EBCDICTB)	A THRU I
	ORG	EBCDICTB+C'J'	UP TO J
	DC	09AL1(*-EBCDICTB)	J THRU R
	ORG	EBCDICTB+C'S'	UP TO S
	DC	08AL1(*-EBCDICTB)	S THRU Z
	ORG	EBCDICTB+C'0'	UP TO 0
	DC	10AL1(*-EBCDICTB)	0 THRU 9
	ORG	EBCDICTB+X'4A'	UP TO CENT SIGN
	DC	07AL1(*-EBCDICTB)	
	ORG	EBCDICTB+C'!'	UP TO EXCLAMATION POINT
	DC	08AL1(*-EBCDICTB)	
	ORG	EBCDICTB+C','	UP TO COMMA
	DC	05AL1(*-EBCDICTB)	
	ORG	EBCDICTB+C':'	UP TO COLON
	DC	06AL1(*-EBCDICTB)	
	END		

Figure 12. Source Code for CSECT COMPAREE

Appendix B. CPXIFACE

Other Interfaces

```
*=====*
```

```
* BELOW IS WHERE THE USER CAN TAILOR THE SOURCE FOR THE SHOP. *
```

```
*=====*
```

```
...
```

&OTH	SETB	1	(YES)	GENERATE THE 'OTHER' INTERFACE.
&ADABAS	SETB	0	(NO)	.ADABAS from Software AG
&CAMLIB	SETB	0	(NO)	.CONDOR CAMLIB
&DATACOM	SETB	0	(FUTURE)	.DATACOMM - Applied Data Research
&DB2	SETB	0	(FUTURE)	.DB2 from IBM (Similar to SQL)
&DL1	SETB	0	(NO)	.DL/1 of IMS or CICS
&DL1FP	SETB	0	(NO)	..IMS'S FASTPATH
&DMS	SETB	0	(NO)	.DMS from Sterling Software
&DOSLB1	SETB	0	(NO)	.DOS Library Structure DASDFP 1.x
&DOSLB2	SETB	0	(NO)	.DOS Library Structure 2.1 and up
&FILEDEF	SETB	0	(NO)	.FILEDEF For CMS - OS Parallel
&FOCUS	SETB	0	(FUTURE)	.FOCUS from Information Builders
&IAM	SETB	0	(NO)	.Innovation Access Method
&ICCF	SETB	0	(NO)	.I.C.C.F. (DOS/VSE)
&IDMS	SETB	0	(NO)	.IDMS from Cullinet
&MODL204	SETB	0	(FUTURE)	.MODEL 204 from C.C.A.
&OWL	SETB	0	(FUTURE)	.Online Without Limits - Pansophic
&PANEXEC	SETB	0	(FUTURE)	.PANEXEC from Pansophic
&POWER	SETB	0	(NO)	.POWER Queues (DOS/VSE)
&RAMIS	SETB	0	(NO)	.RAMIS-II from Mathematica
&ROSCOE	SETB	0	(NO)	.ROSCOE from Applied Data Research
&SPFPAKD	SETB	0	(NO)	.ISPF 2.1 Packed Files (OS Only)
&SPRI	SETB	0	(NO)	.SPRI Queues from Software Pursuits
&SQL	SETB	0	(FUTURE)	.Structured Query Language (VM/DOS)
&SYS2000	SETB	0	(FUTURE)	.SYSTEM 2000 from Intel
&UFAM	SETB	0	(NO)	.UFAM from Citizens Software
&WYLBUR	SETB	0	(NO)	.WYLBUR Format (OS Seq. only)
&ZROLOWN	SETB	0	(NO)	.ROLL YOUR OWN - Proprietary
&ZASCII	SETB	0	(NO)	..ASCII to EBCDIC Translation

```
*=====*
```

```
* ABOVE IS WHERE THE USER CAN TAILOR THE SOURCE FOR THE SHOP. *
```

```
*=====*
```

Figure 13. List of OTHER Interfaces

Assembly

```

//jobname JOB (account),'ASM CPXIFACE',          <=== Change Accordingly
//          CLASS=?,NOTIFY=?,                    <=== Change Accordingly
//          MSGCLASS=?                            <=== Change Accordingly
//*****
//* This is a sample Assembly of 'CPXIFACE'. You may prefer to use *
//* one of your standard procedures such as 'ASMFCL' or 'ASMHCL'   *
//* instead. Just be sure that it assembles clean and link-edits  *
//* without any UNRESOLVED EXTERNAL REFERENCES.                    *
//* -----                                                        *
//*****
//ASMLINK PROC ASM=IEV90,          <==== Name of your Assembler (IFOX00)
//          IEWL=IEWL,            <==== Name of your Link Editor
//          UNIT=SYSDA,           <==== Disk unit type
//          SOMNODE='SYS2.' <==== Highest level node
//ASM      EXEC PGM=&ASM,REGION=512K,
//          PARM='LIST,XREF,RENT,?????' <=== Change Accordingly
//****    IEV90 would probably use PARM='LIST,XREF,RENT,OBJECT'
//****    IFOX00 would probably use PARM='LIST,XREF,RENT,OBJ' and
//****          DDname //SYSGO instead of //SYSLIN
//****
//SYSLIB   DD DISP=SHR,DSN=SYS1.MACLIB
//SYSPRINT DD SYSOUT=*
//SYSPUNCH DD DUMMY
//SYSUT1   DD UNIT=&UNIT,SPACE=(CYL,(1,1))
//SYSLIN   DD DISP=(,PASS),DSN=&&OBJECT,
//          UNIT=&UNIT,SPACE=(TRK,(2,1)),
//          DCB=(RECFM=F,BLKSIZE=80)
//SYSIN    DD DISP=SHR,DSN=&SOMNODE.COMPAREX.IFACE(CPXIFACE)
//*
//LKED     EXEC PGM=&IEWL,PARM='LIST,XREF,MAP,REUS'
//SYSPRINT DD SYSOUT=*
//SYSLIB   DD DISP=SHR,DSN=          <==== System Libraries for:
//*          PANVALET, LIBRARIAN, and/or OTHER Interface
//SYSLMOD  DD DISP=SHR,
//          DSN=&SOMNODE.COMPAREX.loadlib <==== Change Accordingly
//SYSUT1   DD UNIT=&UNIT,SPACE=(CYL,1)
//SYSLIN   DD DISP=(OLD,DELETE),DSN=&&OBJECT
//          DD DDNAME=SYSIN
//          PEND

```

[Figure continued on next page]

[Figure continuation]

```
// *
//ASMLINK EXEC ASMLINK
//LKED.SYSIN DD *
    INCLUDE SYSLIB(PAM)          <==== PANVALET INCLUDE
    INCLUDE SYSLIB(FAIROPN)      <==== LIBRARIAN INCLUDE (OPEN)
    INCLUDE SYSLIB(FAIRMOD)      <==== LIBRARIAN INCLUDE (SEARCH)
    INCLUDE SYSLIB(FAIRREC)      <==== LIBRARIAN INCLUDE (READ)
    INCLUDE SYSLIB(FAIRCLS)      <==== LIBRARIAN INCLUDE (CLOSE)
    ENTRY CPXIFACE
    NAME CPXIFACE(R)
// * EOJ
```

Figure 14. CPXIFASM in somnode.COMPAREX.IFACE

Appendix C. ISPF Related Members

```
//jobname JOB (account),'COMPAREX/ISPF', <=== Change Accordingly
//          CLASS=?,NOTIFY=?, <=== Change Accordingly
//          MSGCLASS=? <=== Change Accordingly
//*****
//* This procedure will unload datasets from the installation *
//* tape which are required for the ISPF interface. *
//* 1: The Jobcard images need updating. *
//* 2: The PROC statement needs updating. *
//* a) TAPEVOL - Installation tape volume serial number *
//* b) DISKVOL - Disk volume serial for target libraries *
//* c) UNIT - Disk unit type *
//* e) SOMNODE - Highest level node *
//* 3: The label parameter assumes that UCC1/TMS is *
//* installed - Remove 'EXPDT=98000' if it is not. *
//*****
//CPXISPF PROC TAPEVOL=?????, <==== Installation tape volume serial
//          DISKVOL=?????, <==== Disk VOLSER for target library
//          UNIT=SYSDA, <==== Disk unit type
//          SOMNODE='SYS2.' <==== Highest level node
//ISPFLIBS EXEC PGM=IEBCOPY,REGION=256K
//SYSPRINT DD SYSOUT=*
//IPANELS DD DISP=SHR,DSN=COMPAREX.PANELS,
//***** DCB=(RECFM=VS,BLKSIZE=6020), <===Invoke if needed
//          VOL=SER=&TAPEVOL,UNIT=TAPE,LABEL=(3,SL,EXPDT=98000)
//IMSGS DD DISP=SHR,DSN=COMPAREX.MSGS,
//          VOL=SER=&TAPEVOL,UNIT=TAPE,LABEL=(4,SL,EXPDT=98000)
//ICLIST DD DISP=SHR,DSN=COMPAREX.CLIST,
//          VOL=SER=&TAPEVOL,UNIT=TAPE,LABEL=(5,SL,EXPDT=98000)
//ISYSGEN DD DISP=SHR,DSN=COMPAREX.SYSGEN,
//          VOL=SER=&TAPEVOL,UNIT=TAPE,LABEL=(6,SL,EXPDT=98000)
//I$ISPF DD DISP=SHR,DSN=COMPAREX.CPX$ISPF,
//          VOL=SER=&TAPEVOL,UNIT=TAPE,LABEL=(7,SL,EXPDT=98000)
//OPANELS DD DISP=(,CATLG,DELETE),DSN=&SOMNODE.COMPAREX.PANELS,
//          UNIT=&UNIT,VOL=SER=&DISKVOL,SPACE=(CYL,(1,1,20))
//OMSGS DD DISP=(,CATLG,DELETE),DSN=&SOMNODE.COMPAREX.MSGS,
//          UNIT=&UNIT,VOL=SER=&DISKVOL,SPACE=(TRK,(1,1,20))
//OCLIST DD DISP=(,CATLG,DELETE),DSN=&SOMNODE.COMPAREX.CLIST,
//          UNIT=&UNIT,VOL=SER=&DISKVOL,SPACE=(TRK,(1,1,20))
//OSYSGEN DD DISP=(,CATLG,DELETE),DSN=&SOMNODE.COMPAREX.SYSGEN,
//          UNIT=&UNIT,VOL=SER=&DISKVOL,SPACE=(TRK,(1,1,20))
//O$ISPF DD DISP=(,CATLG,DELETE),DSN=&SOMNODE.COMPAREX.CPX$ISPF,
//          UNIT=&UNIT,VOL=SER=&DISKVOL,SPACE=(TRK,(1,1,20))
//          PEND
```

[Figure continued on next page]

[Figure continuation]

```
//*  
//ISPF      EXEC    CPXISPF  
            COPY    I=IPANELS,O=OPANELS  
            COPY    I=IMSGS,O=OMSGS  
            COPY    I=ICLIST,O=OCLIST  
            COPY    I=ISYSGEN,O=OSYSGEN  
            COPY    I=ISISPF,O=OSISPF  
  
//* EOJ
```

Figure 15. CPXISPF in somnode.COMPAREX.IFACE

Link CPX\$ISPF

```
//jobname JOB (account),'LINK CPX$ISPF',      <=== Change Accordingly  
//              CLASS=?,NOTIFY=?,              <=== Change Accordingly  
//              MSGCLASS=?                      <=== Change Accordingly  
//*****  
//* This procedure will Linkage Edit the object code for module      *  
//* CPX$ISPF into a new library called 'somnode.COMPAREX.LOAD'.      *  
//*****  
//LINKIT  PROC IEWL=IEWL,      <==== Name of your Link Editor  
//              UNIT=SYSDA,      <==== Disk unit type  
//              SOMNODE='SYS2.' <==== Highest level node  
//LKED      EXEC PGM=&IEWL,PARM='LIST,XREF,LET,MAP,RENT'  
//SYSPRINT DD SYSOUT=*  
//**SYSLIB DD DISP=SHR,  
//**              DSN=somnode.ISPLOAD      <=== Where ISPLINK is  
//SYSUT1 DD UNIT=&UNIT,SPACE=(CYL,1)  
//SYSLMOD DD DISP=(NEW,CATLG,DELETE),DSN=&SOMNODE.COMPAREX.LOAD,  
//              UNIT=&UNIT,SPACE=(TRK,(1,1,1)),  
//              DCB=(DSORG=PO,RECFM=U,BLKSIZE=6000)  
//SYSLIN DD DISP=SHR,  
//              DSN=&SOMNODE.COMPAREX.CPX$ISPF(CPX$ISPF)  
//              DD DDNAME=SYSIN  
//              PEND  
//LINKIT  EXEC LINKIT  
//SYSIN DD *  
          ENTRY CPX$ISPF  
          NAME CPX$ISPF(R)
```

Figure 16. LNKISPF in somnode.COMPAREX.IFACE

IBM Model Clist - Allocation for Sysgen

PROC 0

```

/*****
/* IIIIIIII BBBB M      M AAAAA L      1      */
/* I      B      B MM    MM A      A L      11     */
/* I      B      B M M M M A      A L      1      */
/* I      BBBB M      M M AAAAAA L      1      */
/* I      B      B M      M A      A L      1      */
/* I      B      B M      M A      A L      1      */
/* IIIIIIII BBBB M      M A      A LLLLLL 11111    */
/*                                                     ****/
/*                                                     IBM Users Only */
/*                                                     ****/
/* NOTE - Set "CAPS ON" when you edit this!          */
/*****/

FREE FI(ISPPLIB,ISPTLIB)
/*****/
/* Always concatenate the higher block size first! */
/*****/
ALLOC FI(ISPPLIB) DA('somnode.COMPAREX.PANELS'      +
                    'ISR.V2R2M0.ISRPLIB'            +
                    'ISP.V2R2M0.ISPPLIB')            SHR
ALLOC FI(ISPTLIB) DA('somnode.COMPAREX.TABLES'      +
                    'ISR.V2R2M0.ISRTLIB'            +
                    'ISP.V2R2M0.ISPTLIB')            SHR

/*****/
/* No concatenations allowed for 'ISPTABL'!          */
/*****/

FREE FI(ISPTABL)
ALLOC FI(ISPTABL) DA('somnode.COMPAREX.TABLES')      SHR

```

Figure 17. IBMAL1 in somnode.COMPAREX.CLIST

IBM Model Clist - Normal Allocation

```

PROC 0 +
  LIST CONLIST SYMLIST

/*****
/* IIIIIIII BBBB M      M  AAAA  L      22222      */
/*      I      B      B MM  MM A      A L      2      2      */
/*      I      B      B M M M A      A L      2      */
/*      I      BBBB M  M  M  AAAAAA L      2      */
/*      I      B      B M      M A      A L      2      */
/*      I      B      B M      M A      A L      2      */
/* IIIIIIII BBBB M      M A      A LLLLLL 222222      */
/*                                                              ****/
/*                                                              IBM Users Only */
/*                                                              ****/
/* NOTE - Set "CAPS ON" when you edit this!      */
/*****/

IF &LIST> THEN CONTROL LIST
IF &CONLIST> THEN CONTROL CONLIST
IF &SYMLIST> THEN CONTROL SYMLIST

CONTROL NOMSG
FREE FI(ISPPLIB,ISPMLIB,ISPLLIB,ISPTLIB)
CONTROL MSG

/*****
/* Always concatenate the higher block size first! */
/*****/

ALLOC FI(ISPPLIB) DA('somnode.COMPAREX.PANELS'      +
                    'ISR.V2R2M0.ISRPLIB'            +
                    'ISP.V2R2M0.ISPPLIB')            SHR
ALLOC FI(ISPTLIB) DA('somnode.COMPAREX.TABLES'      +
                    'ISR.V2R2M0.ISRTLIB'            +
                    'ISP.V2R2M0.ISPTLIB')            SHR
ALLOC FI(ISPMLIB) DA('somnode.COMPAREX.MSGS'        +
                    'ISR.V2R2M0.ISRMLIB'            +
                    'ISP.V2R2M0.ISPMLIB')            SHR
ALLOC FI(ISPLLIB) DA('somnode.COMPAREX.LOAD'        +
                    'ISR.V2R2M0.ISRLOAD'            +
                    'ISP.V2R2M0.ISPLOAD')            SHR

```

Figure 18. IBM4L2 in somnode.COMPAREX.CLIST

COMPAREX[®] 6.1.0 (87/054)

Quick

Start

OS

Marketing Representative:

STERLING SOFTWARE
Systems Software Marketing Division
11050 White Rock Road
Rancho Cordova, California 95670-6095
Phone: (916) 635-5535
Telex: 377314

Copyright © SERENA Consulting 1987.
Licensed Material.

COMPAREX[®] is a Registered Trademark of:

SERENA Consulting
Post Office Box 117039
Burlingame, California 94010
Phone: (415) 347-0100
Telex: 172040 HQ SMT

Table of Contents

Preface	1
Version, Release, Modification	1
Purpose	2
Chapter 1 - Getting Started	3
General Flowchart	3
Create Two Known Files	4
Chapter 2 - Data Files	5
DATA Comparison - Mostly Defaults	5
DATA Comparison - Key Synchronization	7
Differences Report with DATA	9
Chapter 3 - Text Files	10
Differences Report with TEXT	11
Differences Between DATA and TEXT	11
DATA or TEXT	12
Chapter 4 - Interfaces	13
PANVALET, LIBRARIAN, and GEM	13
ISPF Interface	16
Connected to Main Menu	16
DATA Comparison Via Short Cut	17
Index	20

List of Illustrations

Figure 1. COMPAREX General Flowchart	3
Figure 2. Sample Files - FILE1 And FILE2	4
Figure 3. Data File Comparison	5
Figure 4. Differences Report - MAXDIFF Only	7
Figure 5. Data File Comparison - Key and Mask	8
Figure 6. Differences Report - DATA, KEY, FORMAT, MASK	9
Figure 7. Text File Comparison	10
Figure 8. Differences Report - TEXT = JCL	11
Figure 9. Direct PANVALET Interface	13
Figure 10. Direct LIBRARIAN Interface	14
Figure 11. Direct LIBRARIAN Interface - Directories	14
Figure 12. Direct GEM Interface	15
Figure 13. ISPF Panel VENDOR - Raw Form	16
Figure 14. ISPF Panel CPX@PRIM	17
Figure 15. ISPF Panel CPXSHORT	18
Figure 16. ISPF Panel CPXJOBST	19

Preface

COMPAREX runs under the three major IBM operating systems:

- OS/VS - (MVS, MVS/XA)
- DOS/VSE - (SP 2.1 System Package, and pre-2.1)
- VM/CMS

In addition, it runs under these operating systems:

- MVT/VSE from Software Pursuits
- OSIV/F4 from FUJITSU/FACOM

Version, Release, Modification

All software created and maintained by SERENA Consulting will have a Version, Release, and Modification level associated with it. Only when the Version or Release number changes (*usually annually*) will the full customer base (*those who are up to date with maintenance*) be issued new tapes and documentation.

There will be at least two interim Modifications during the year. At year end, the accumulated Modifications will be rolled into the next general Release.

This manual describes:

COMPAREX OS
Version 6
Release 1
Modification 0
Julian release date 87/054

This COMPAREX 6.1.0 Quick Start Manual is a reference document that gives the information most needed by COMPAREX users. It illustrates, by example, how to invoke COMPAREX for solving some of the most common comparison situations.

The COMPAREX 6.1.0 User Reference Manual gives an extensive discussion of the functions and processing logic of COMPAREX, the comparison utility. The User Reference Manual has been written for the professional programmer who has experience with programming, file structures and organizations, utilities, and testing practices. The manual contains extensive information about the COMPAREX keywords and messages.

Finally, the COMPAREX 6.1.0 User Reference Card shows the format and the description of each keyword.

Purpose

The purpose of this Quick Start Manual is to:

- enable you to use COMPAREX easily and quickly
- guide you through some commonly used comparisons
- discuss the Differences Report and its parts
- discuss Data logic versus Text logic

Step by step instructions will show you how to:

- compare files as a submitted batch job
- compare files in foreground using the ISPF Interface

Chapter 1 - Getting Started

This is only a Quick Start, a supplement to the User Reference Manual which covers COMPAREX applications in more detail. You are urged to read the User Reference Manual in full.

If you have any problem relating to COMPAREX, call your marketing representative as described on the initial page of this document.

Note: If you have not yet installed COMPAREX, please refer to the installation guide that accompanied your tape.

General Flowchart

"COMPAREX General Flowchart" below shows the five COMPAREX files.

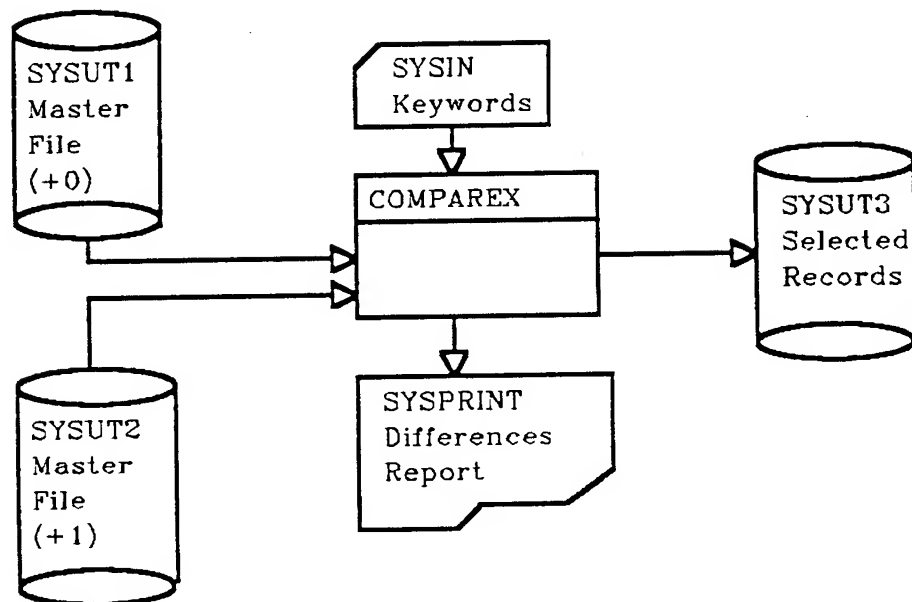


Figure 1. COMPAREX General Flowchart

INPUT - three files are used:

- SYSIN contains any keywords to modify the all-defaults mode. If SYSIN is not able to be opened or if SYSIN is empty, COMPAREX processes without any keywords, taking all defaults. If SYSIN is not empty, COMPAREX modifies its default processing with the user's keywords.

- SYSUT1 is the first file of the two to be compared. Usually, this is the baseline file or old master. If program source code is being compared, this is the unmodified version.
- SYSUT2 is the second file of the two to be compared. Usually, this is the testline file or new master. If program source code is being compared, this is the updated version.
- If PANVALET, LIBRARIAN, GEM, or OTHER file structures are to be processed directly, the proper ddname must be supplied instead of SYSUT1 or SYSUT2 or both so that the COMPAREX interface (*CPXIFACE*) can access it.

OUTPUT - two files are generated:

- SYSPRINT contains the Differences Report, a listing of the results of the comparison. It also contains a list of the defaults and keywords used, and it shows end-of-job statistics lines. The user can also specify a list of the COMPAREX keywords and a visual representation of each record type identified.
- SYSUT3 is an optional file. If present, it contains differing records from file SYSUT2. COMPAREX can be used as a test file generator by the specification of COPYDIFF (*the keyword that directs COMPAREX to write file SYSUT3*).

Create Two Known Files

In order to do simple comparisons with known data, allocate and create two datasets (named FILE1 and FILE2) as in "Sample Files - FILE1 And FILE2" below.

somnode.FILE1

```
RECORD01AAAAAAAAAAAA
RECORD02BBBBBBBBBBBBB
RECORD03CCCCCCCCCCCCC
RECORD04DDDDDDDDDDDDD
RECORD05EEEEEEEEEEEEEE
RECORD06FFFFFFFFFFFFFFF
RECORD07GGGGGGGGGGGGG
RECORD08HHHHHHHHHHHHH
RECORD09IIIIIIIIIIIII
RECORD10JJJJJJJJJJJJJ
RECORD26 INSERTION
```

somnode.FILE2

```
RECORD01AAAAAAAAAAAA
RECORD02BBBBBBBBBBBBB
RECORD03FIRST CHANGE
RECORD04DDDDDDDDDDDDD
RECORD05EEEEEEEEEEEEEE
RECORD06SECONDCHANGE
RECORD07GGGGGGGGGGGGG
RECORD08HHHHHHHHHHHHH
RECORD09IIIIIIIIIIIII
RECORDXX INSERTION
RECORD10JJJJJJJJJJJJJ
RECORD11 INSERTION
```

Figure 2. Sample Files - FILE1 And FILE2

Chapter 2 - Data Files

The examples given are in the simplest form, using as few keywords as possible. We recommend that you try these before you proceed to tailor COMPAREX to your specific needs.

If you need keywords explained, refer to the **User Reference Manual** or the **User Reference Card**.

DATA Comparison - Mostly Defaults

"Data File Comparison" below will produce a simple DATA comparison, pairing physical record to physical record without any attempt to match on keys. If you use this basic comparison, as soon as an out-of-sync situation is found, COMPAREX will print all remaining records as differing. Therefore, we have included:

MAXDIFF=nn where nn is a small number to limit your print output

CONTINUE to enable you to print out the end of job statistics line showing the number of differences found.

```
//OS$JOB    JOB ...
//COMPARE  PROC
//CPX      EXEC PGM=COMPAREX,REGION=200K
//STEPLIB   DD DISP=SHR,DSN=somnode.COMPAREX.LOADLIB
//SYSPRINT  DD SYSOUT=*
//          PEND
//*
//DOIT     EXEC COMPARE
//SYSUT1    DD DISP=SHR,DSN=somnode.FILE1
//SYSUT2    DD DISP=SHR,DSN=somnode.FILE2
//SYSIN     DD *
    MAXDIFF=5    /* to avoid a runaway compare */
    CONTINUE     /* to see final count of differences */
/*
```

Figure 3. Data File Comparison

Submit the job in "Data File Comparison" on page 5. Your output should look something like this:

C O M P A R E X (OS-6.1.0 - 87/054)

MONDAY FEBRUARY 23, 1987 (87/054) 17:11:59 PAGE 1

PROPRIETARY SOFTWARE PRODUCT OF SERENA CONSULTING PHONE (415)347-0100 OR TELEX 172040 HQ SMT

LICENSED TO: your corporate name/trial offer expires ...
your city, state, zip/agent contact for license ...

ALL OTHER RIGHTS RESERVED - USE OF THIS SOFTWARE
PRODUCT BY UNAUTHORIZED PERSONS IS PROHIBITED.

```
CPX001 - *****
CPX001 - *** I N S T A L L A T I O N   D E F A U L T S ***
CPX001 - *****
CPX001 - HALT=COND /* STOP EXECUTION IF SYNTAX ERRORS FOUND */
CPX001 - *****
CPX001 - *** END OF INSTALLATION DEFAULTS ***
CPX001 - *****
CPX001 - MAXDIFF=5 /* to avoid a runaway compare */
CPX001 - CONTINUE /* to see final count of differences */

CPX031 - EXECUTION OF OSSJOB.CPX.DOIT - VALUES EXTRACTED/DEFAULTED:
CPX041 - MAXDIFF=5,CONTINUE,STOPAFT=999999999999
CPX051 - PRINT=(MATCH,MISMATCH),MBRHOR=YES,HALT=COND
CPX061 - WILDCARD=C'.' ,MODE=APPLICATIONS,(ALL DISPLACEMENTS RELATIVE TO ONE)
CPX081 - DECIMAL,EBCDIC,CASE=UPPER,LINE=(32,HORIZONTAL),PAGE=58
CPX111 - DASH=C'-' ,PLUS=C'+'
CPX211 - SYSUT1=somnode.FILE1 DCB=(DSORG=PS,RECFM=F,BLKSIZE=80)
CPX221 - SYSUT2=somnode.FILE2 DCB=(DSORG=PS,RECFM=F,BLKSIZE=80)
CPX251 - DATA,FORMAT=02,INTERLEAVE=0
        (FORMAT EXPLANATION: FULL SYSUT1 FOLLOWED BY DIFFERING LINES OF SYSUT2)
```

C O M P A R E X (OS-6.1.0 - 87/054)

MONDAY FEBRUARY 23, 1987 (87/054) 17:11:59 PAGE 2

SYSUT1=somnode.FILE1,SYSUT2=somnode.FILE2

CPX511 - RECORD NUMBER 3 ON FILE SYSUT1

```
1    D9C5C3D6 D9C4F0F3 C3C3C3C3 C3C3C3C3 C3C3C3C3 40404040 ... *RECORD03CCCCCCCCCCCC * O N E
33   40404040 40404040 40404040 40404040 40404040 40404040 ... *
65   40404040 40404040 40404040 40404040 40404040 40404040 ... *
```

CPX521 - RECORD NUMBER 3 ON FILE SYSUT2

```
1    D9C5C3D6 D9C4F0F3 C6C9D9E2 E340C3C8 C105C7C5 40404040 ... *RECORD03FIRST CHANGE * T W O
      .....          .....          .....          .....          .....
                                         -DIFFERENCE+
```

(We will edit out record number 6 "SECONDCHANGE"
because it is similar.)

CPX511 - RECORD NUMBER 10 ON FILE SYSUT1

```
1    D9C5C3D6 D9C4F1F0 D1D1D1D1 D1D1D1D1 D1D1D1D1 40404040 ... *RECORD10JJJJJJJJJJJJ * O N E
33   40404040 40404040 40404040 40404040 40404040 40404040 ... *
65   40404040 40404040 40404040 40404040 40404040 40404040 ... *
```

CPX521 - RECORD NUMBER 10 ON FILE SYSUT2

```
1    D9C5C3D6 D9C4E7E7 40C9D5E2 C5D9E3C9 D6D54040 40404040 ... *RECORDXX INSERTION * T W O
      .....          .....          .....          .....          .....
                                         -DIFFERENCE+
```

(Record 11 edited out.)

CPX711 - END OF DATA ON FILE SYSUT1

CPX571 - EXTRA RECORD NUMBER 12 ON FILE SYSUT2

```

(Record 12 from file SYSUT2 displayed here.)
CPX67I - MAXDIFF INVOKED, CONTINUING WITHOUT PRINTING BY REQUEST
CPX72I - END OF DATA ON FILE SYSUT2
CPX74I - BYTES UNDERSCORED(51)
CPX75I - RECORDS PROCESSED: SYSUT1(11)/SYSUT2(12),DIFFERENCES(4,0,1)
          EXPLANATION - 4 RECORDS DIFFER THAT SYNCHRONIZED TOGETHER
                        0 RECORDS WERE CONSIDERED INSERTED ON SYSUT1
                        1 RECORD WAS CONSIDERED INSERTED ON SYSUT2
CPX80I - TIME OF DAY AT END OF JOB: 08:52:34 - CONDITION CODE ON EXIT: 4

```

Figure 4. Differences Report - MAXDIFF Only

The format of the totals for message CPX75I is:

SYSUT1(n1)/SYSUT2(n2)SYSUT3(n3),DIFFERENCES(d0,d1,d2) where
 n1 = number of records read from SYSUT1
 n2 = number of records read from SYSUT2
 n3 = number of records written to SYSUT3
 d0 = number of pairs of records which differed
 d1 = number of records on SYSUT1, not on SYSUT2
 d2 = number of records on SYSUT2, not on SYSUT1

If you didn't have any problems, print your output. If you did have problems:

1. . Is your JCL correctly formatted?
2. Are your dataset names spelled correctly?
3. Do the datasets exist?
4. Where there any syntax (message CPX30A) errors?
5. Rerun.

DATA Comparison - Key Synchronization

We will specify the logical key to COMPAREX as being the first eight bytes of each record. It is in character format and ascending. The syntax for the KEY keyword is to specify the displacement (1) and the length (8):

```
KEY=( 1 , 8 )
```

If no key is specified but the file organization uses a key (such as ISAM or VSAM-KSDS), COMPAREX will use that key without you specifying it.

You may have noticed that the Differences Report generated by the previous keywords (*or lack of*) was in the IBM Dump format. A lot of hexadecimal characters were displayed. Since our data did not have any **HEX** characters, this time let's just see the alphanumeric characters. To do this, specify:

```
LINE=80
```

When you created the FILE1 and FILE2 datasets, was there a sequence number in columns 73 through 80? If there was, COMPAREX probably underscored that also in your execution. To avoid this again, specify:

MASK=(73,END)

"Data File Comparison - Key and Mask" below will produce a DATA comparison, pairing records that match on the specified key and isolating inserted and deleted records.

```
//OS$JOB    JOB ...
//COMPARE PROC
//CPX      EXEC PGM=COMPAREX,REGION=200K
//STEPLIB   DD DISP=SHR,DSN=somnode.COMPAREX.LOADLIB
//SYSPRINT  DD SYSOUT=*
//
//          PEND
// *
//DOIT     EXEC COMPARE
//SYSUT1    DD DISP=SHR,DSN=somnode.FILE1
//SYSUT2    DD DISP=SHR,DSN=somnode.FILE2
//SYSIN     DD *
    MAXDIFF=5      /* to avoid a runaway compare */
    CONTINUE       /* to see final count of differences */
    LINE=80        /* Format in alphanumeric only - no HEX characters */
    MASK=(73,END)  /* Ignore any sequence number in columns 73 - 80 */
    KEY=(1,8)      /* KEY will be the field in first 8 characters */
/*
```

Figure 5. Data File Comparison - Key and Mask

Submit the job in "Data File Comparison - Key and Mask" on page 8. Your output should look something like "Differences Report - DATA, KEY, FORMAT, MASK" on page 9.

Through the use of KEYs, this run of COMPAREX will detect an out-of-sync situation. These records that are determined to be out-of-sync will be written to the Differences Report. The output will differ from the previous run because COMPAREX will now resynchronize after insertions and deletions.

```
DSPL |...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8

CPX51I - RECORD NUMBER 3 ON FILE SYSUT1
1    RECORD03CCCCCCCCCCCCC                                O N E

CPX52I - RECORD NUMBER 3 ON FILE SYSUT2    FIELD=1
1    RECORD03FIRST CHANGE                                T W O
        ..... -DIFFERENCE+

CPX51I - RECORD NUMBER 6 ON FILE SYSUT1
1    RECORD06FFFFFFFFFFFFF                                O N E

CPX52I - RECORD NUMBER 6 ON FILE SYSUT2    FIELD=1
1    RECORD06SECONDCHANGE                                T W O
        ..... -DIFFERENCE+

CPX36A - KEY OUT OF SPECIFIED SEQUENCE - RECORD 10 ON FILE SYSUT2
1    RECORDXX INSERTION                                T W O

CPX62I - KEY SYNCHRONIZATION MISMATCH - RECORD 10 ON FILE SYSUT2
1    RECORDXX INSERTION                                T W O

CPX62I - KEY SYNCHRONIZATION MISMATCH - RECORD 12 ON FILE SYSUT2
1    RECORD11 INSERTION                                T W O

CPX72I - END OF DATA ON FILE SYSUT2

CPX61I - KEY SYNCHRONIZATION MISMATCH - RECORD 11 ON FILE SYSUT1
1    RECORD26 INSERTION                                O N E

CPX67I - MAXDIFF INVOKED, CONTINUING WITHOUT PRINTING BY REQUEST

CPX71I - END OF DATA ON FILE SYSUT1

CPX74I - BYTES UNDERSCORED(23)

CPX75I - RECORDS PROCESSED: SYSUT1(11)/SYSUT2(12),DIFFERENCES(2,1,2)
        EXPLANATION - 2 RECORDS DIFFER THAT SYNCHRONIZED TOGETHER
                      1 RECORD WAS CONSIDERED INSERTED ON SYSUT1
                      2 RECORDS WERE CONSIDERED INSERTED ON SYSUT2

CPX80I - TIME OF DAY AT END OF JOB: 18:47:28 - CONDITION CODE ON EXIT: 4
```

Figure 6. Differences Report - DATA, KEY, FORMAT, MASK

Differences Report with DATA

Differing records are displayed as follows:

- The literal ONE is shown in the right-hand column when the line contains data from a record from SYSUT1.
- The literal TWO is shown in the right-hand column when the line contains data from a record from SYSUT2.
- The literal DIFFERENCE is shown in the right-hand column when the SYSUT2 record has been selected for printing because it differs from the record on SYSUT1.
- The amount of detail of files and differing records depends on the FORMAT keyword.

In this example, the format used is "full SYSUT1 followed by differing lines of SYSUT2."

Chapter 3 - Text Files

We will use Text logic to compare the same two files. Assume that there is a sequence number in columns 73 through 80. Therefore

TEXT=JCL

will ignore the sequence number and use the first 72 columns for the comparison.

```
//OS$JOB    JOB ...
//COMPARE  PROC
//CPX      EXEC  PGM=COMPAREX,REGION=200K
//STEPLIB  DD  DISP=SHR,DSN=somnode.COMPAREX.LOADLIB
//SYSPRINT DD  SYSOUT=*
//          PEND
//*
//TEXT     EXEC  COMPARE
//SYSUT1   DD  DISP=SHR,DSN=somnode.FILE1
//SYSUT2   DD  DISP=SHR,DSN=somnode.FILE2
//SYSIN    DD  *
    MAXDIFF=5    /* to avoid a runaway compare */
    CONTINUE     /* to see final count of differences */
    TEXT=JCL     /* Compare contents of first 72 bytes only */
/*
```

Figure 7. Text File Comparison

Submit the job in "Text File Comparison" on page 10. Your output should look something like "Differences Report - TEXT=JCL" on page 11. Rather than underscoring the differences, COMPAREX frames in a TEXT comparison.

```

+++++<++++.++++1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7>++++.++++8+++++
+ RECORD03CCCCCCCCCCCC DIF ONE 3 +
+ |-----1-----2-----3-----4-----5-----6-----7-----8-----+
+ RECORD03FIRST CHANGE DIF TWO 3 +
+++++<++++.++++1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7>++++.++++8+++++

+++++<++++.++++1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7>++++.++++8+++++
+ RECORD06FFFFFFFFFFFF DIF ONE 6 +
+ |-----1-----2-----3-----4-----5-----6-----7-----8-----+
+ RECORD06SECONDCHANGE DIF TWO 6 +
+++++<++++.++++1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7>++++.++++8+++++

+++++<++++.++++1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7>++++.++++8+++++
+ RECORDXX INSERTION DIF TWO 10 +
+++++<++++.++++1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7>++++.++++8+++++

+++++<++++.++++1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7>++++.++++8+++++
+ RECORD26 INSERTION DIF ONE 11 +
+ |-----1-----2-----3-----4-----5-----6-----7-----8-----+
+ RECORD11 INSERTION DIF TWO 12 +
+++++<++++.++++1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7>++++.++++8+++++

CPX71I - END OF TEXT ON FILE SYSUT1
CPX72I - END OF TEXT ON FILE SYSUT2
CPX75I - RECORDS PROCESSED: SYSUT1(11)/SYSUT2(12),DIFFERENCES(3,0,1)
        EXPLANATION - 3 RECORDS DIFFER THAT SYNCHRONIZED TOGETHER
                      0 RECORDS WERE CONSIDERED INSERTED ON SYSUT1
                      1 RECORD WAS CONSIDERED INSERTED ON SYSUT2

CPX80I - TIME OF DAY AT END OF JOB: 08:52:39 - CONDITION CODE ON EXIT: 4

```

Figure 8. Differences Report - TEXT=JCL

Differences Report with TEXT

Since COMPAREX compares blocks of significant data rather than byte by byte using TEXT Processing logic, an entire record is displayed without underscoring differing bytes or half-bytes or excess bytes.

Differing records/blocks are shown as follows:

- records from SYSUT1 not matched with records from SYSUT2 are identified with the designation DIF ONE on the right-hand side of the report. Similarly records from SYSUT2 which are not matched are identified with DIF TWO.
- The logical record number is shown on the far right.
- When PRINT=FULL is specified, all records are shown in context.
- If FRAME is used, the differing records are FRAMED with the DASH and PLUS characters.

Differences Between DATA and TEXT

COMPAREX recognizes two categories of files: DATA and TEXT. There are separate logic routines for each, since they differ in their means of synchronization.

DATA is defined, in COMPAREX, as any file in which there is a known inter-record relationship. DATA files have bytes and fields in fixed positions on each record.

Using DATA file comparison logic, files may be compared record to record, using 'same physical number' synchronization. For example, record1 to record1, record100 to record100, etc. If neither DATA nor TEXT is specified, this is the default. Synchronization may also be performed using KEY and SEGMENT.

Examples of DATA files are:

- master files
- intermediate files
- transaction files
- databases
- load modules

TEXT is defined, in COMPAREX, as any file in which there is not a known inter-record relationship. TEXT files do not have bytes and fields in fixed positions; records may contain blanks and may be entirely free-form. Rather than pairing records for comparison, TEXT synchronization uses buffers and look-ahead logic to pair blocks and isolate differences.

Examples of TEXT files are:

- program source code
- JCL
- reports
- documentation (such as this manual)

DATA or TEXT

If files have keys, use DATA with KEY synchronization.

If files are databases, use DATA with KEY or SEGMENT synchronization.

If files are object code, use DATA without any key.

If datasets are load modules, use the CSECT Parsing option to DATA:

DATA=CSECT

Otherwise, use TEXT, increasing BUFF and lowering MLC until the desired results are obtained.

This manual fulfilled its purpose if you successfully ran:

- DATA comparison without keys
- DATA comparison with keys
- TEXT comparison

and, you were able to understand the printed output of each.

Chapter 4 - Interfaces

COMPAREX interfaces to many (via CPXIFACE) different data collection structures such as PANVALET, LIBRARIAN, DL/1, IDMS etc. The success of these direct interfaces is very often dependent on how CPXIFACE was generated. There is a friendly interface from the user to COMPAREX called the ISPF interface.

PANVALET, LIBRARIAN, and GEM

COMPAREX can interface directly to Pansophic's PANVALET, Applied Data Research's LIBRARIAN, or FUJITSU/FACOM's GEM.

Refer to "Direct PANVALET Interface" below for an example of comparing two members on the same PANVALET library and generating an audit trail for subsequent browsing and/or feeding back into PAN#1.

```
//COMPARE EXEC PGM=COMPAREX,REGION=512K
//STEPLIB DD DISP=SHR,DSN=somnode.COMPAREX.LOADLIB
//SYSPRINT DD SYSOUT=*
//PANDD1 DD DISP=SHR,DSN=somnode.PANLIB
//SYSUT3 DD DISP=(,PASS),DSN=&&AUDIT,
// UNIT=SYSDA,SPACE=(TRK,(1,1)),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//SYSIN DD *
    SYSUT1=(PAN,MEMBER=cobolname1) /* Fill in cobolname1 */
    SYSUT2=(PAN,MEMBER=cobolname2) /* Fill in cobolname2 */
    TEXT=COBOL
    COPYDIFF=(PAN,STAMP=YES) /* Generate Audit Trail */
//*
//TRAIL EXEC PAN$PROC <== Invokes PGM=PAN#1
//PANDD1 DD DISP=SHR,DSN=somnode.PANLIB
//SYSIN DD DISP=(OLD,DELETE),DSN=&&AUDIT
//* EOJ
```

Figure 9. Direct PANVALET Interface

See "Direct LIBRARIAN Interface" on page 14 below for an example of comparing a member of a LIBRARIAN Master against a sequential dataset.

Note: Your LIBRARIAN release level must be 3.2 or higher. Release 3.1 will not work at all.

```
//COMPARE EXEC PGM=COMPAREX,REGION=512K
//STEPLIB DD DISP=SHR,DSN=somnode.COMPAREX.LOADLIB
//SYSPRINT DD SYSOUT=*
//MASTER DD DISP=SHR,DSN=somnode.LIBARIAN.MASTER
//SYSUT2 DD DISP=SHR,DSN=somnode.PROGRAM1.ASM
//SYSIN DD *
  SYSUT1=(LIB, MEMBER=PROGRAM1)
  TEXT=BAL
//* EOJ
```

Figure 10. Direct LIBRARIAN Interface

Refer to "Direct LIBRARIAN Interface - Directories" on page 14 for an example of comparing the directories of two LIBRARIAN Masters.

```
//COMPARE EXEC PGM=COMPAREX,REGION=512K
//STEPLIB DD DISP=SHR,DSN=somnode.COMPAREX.LOADLIB
//SYSPRINT DD SYSOUT=*
//LIB1 DD DISP=SHR,DSN=somnode.LIBARIAN.MASTER1
//LIB2 DD DISP=SHR,DSN=somnode.LIBARIAN.MASTER2
//SYSIN DD *
  SYSUT1=(LIB,DDNAME=LIB1),SYSUT2=(LIB,DDNAME=LIB2)
  DIR=PDF /* Compare the directories only */
//* EOJ
```

Figure 11. Direct LIBRARIAN Interface - Directories

Refer to "Direct GEM Interface" on page 15 for an example of comparing a range of members from a GEM library against a PDS.

```

//COMPARE EXEC PGM=COMPAREX,REGION=256K
//STEPLIB DD DISP=SHR,DSN=somnode.COMPAREX.LOADLIB
//SYSPRINT DD SYSOUT=*
//GEM DD DISP=SHR,DSN=somnode.GEM <=== FUJITSU/FACOM Only
//SYSUT2 DD DISP=SHR,DSN=somnode.PDS
//SYSIN DD *
SYSUT1=(LIB,DDNAME=GEM) /* Get to GEM as 'LIB' */
TEXT=$. /* Wildcard TEXT */
FILTERIN=(MEMBER,1,GE,C'CPX') /* Members starting */
FILTERIN=(MEMBER,1,LE,C'FOX') /* with 'CPX' thru 'FOX' */
/* EOJ

```

Figure 12. Direct GEM Interface

ISPF Interface

The ISPF interface consists of:

- Panel library
- Messages library
- Clist library (*strictly for installation*)
- Sysgen library
- Object code library for module CPX\$ISPF

Connected to Main Menu

Depending on how the interface was installed, you could go directly to "ISPF Panel CPX@PRIM" on page 17 or (*as recommended*) through "ISPF Panel VENDOR - Raw Form" below.

```
)BODY width(&zscreenw) expand(!!)
%!!-! Vendor Supplied Software !!-!
%Option ==>_ZCMD      +
%
% 1 +COMPAREX - Invoke%COMPAREX/ISPF
% 2 +CHANGE MAN - Invoke%Change Man+Services
% X +EXIT      -%Exit+to primary option menu
%
+Enter%END+command to return to primary option menu.
%
)PROC
  &ZSEL = TRANS( TRUNC (&ZCMD, '.')
    1, 'PANEL(CPX@PRIM) NEWAPPL(CPX) NEWPOOL'
    2, 'CMD(CMNINIT) NEWAPPL(CMN) NEWPOOL'
    , , ,
    X, 'EXIT'
    *, '?' )
)END
```

Figure 13. ISPF Panel VENDOR - Raw Form

The interface is panel driven. From your main menu, (ISR@PRIM), you eventually get to the COMPAREX primary menu - "ISPF Panel CPX@PRIM" below.

```
..... COMPAREX/ISPF 6.1.0 Primary Menu .....
Option ==>

0 - SHORT CUT  - Single screen for options and dataset names
1 - OPTIONS    - Specify compare options for this session
2 - DS NAMES   - Specify dataset names to be compared
3 - SAVE       - Save options profile for future sessions
4 - LOAD       - Select/Delete options profile from prior session(s)
5 - CLEAR      - Clear previously loaded profile
6 - FOREGROUND - Invoke COMPAREX in the foreground and wait
7 - BACKGROUND - Invoke COMPAREX as a submitted batch job
8 - BACKGROUND - Similar to above but edit job (and optionally SUBMIT)
X - EXIT       - Exit

Press HELP KEY for tutorial assistance at any point;
Enter END Command to exit.
```

Figure 14. ISPF Panel CPX@PRIM

Note that the cursor is positioned at the

```
Option ==> _
```

where you may enter an option (0 to 8 or X) or TSO commands such as:

```
Option ==> TSO LISTC LEVEL(somnode.COMPAREX)
```

or

```
Option ==> TSO STATUS
```

TSO commands may be processed at any point (*except TUTORIAL*) within the COMPAREX interface.

DATA Comparison Via Short Cut

Choose the "Short Cut" method of specifying dataset names and brief options by entering 0 at:

```
Option ==> 0
```

Now, "ISPF Panel CPXSHORT" below will be displayed asking you to specify both file names and the free-form keywords required to compare them.

```

..... Dataset Names and Brief Options .....
Command ==>

SYSUT1 ISPF FILE:                                SYSUT2 ISPF FILE:
PROJECT ==> _____                          PROJECT ==> _____
LIBRARY ==> _____                          LIBRARY ==> _____
TYPE ==> _____                             TYPE ==> _____
MEMBER ==> _____                           MEMBER ==> _____

OTHER PARTITIONED, SEQUENTIAL, VSAM, OR ISAM DATASET:
SYSUT1 DSNAME ==> _____
SYSUT2 DSNAME ==> _____

ENTER FREE FORM KEYWORDS BELOW: (No Syntax Checking Done on the Panel)
==> _____
==> _____
==> _____
==> _____

Press ENTER to register; Enter END Command to exit.

```

Figure 15. ISPF Panel CPXSHORT

We will enter the two dataset names to compare in the OTHER PARTITIONED, SEQUENTIAL, VSAM, OR ISAM DATASET like this:

```

OTHER PARTITIONED, SEQUENTIAL, VSAM, OR ISAM DATASET:
SYSUT1 DSNAME ==> 'somnode.FILE1' _____
SYSUT2 DSNAME ==> 'somnode.FILE2' _____

```

Since by now, we know exactly how to compare these two datasets, we need only say:

```

ENTER FREE FORM KEYWORDS BELOW: (No Syntax Checking ...
==> data,key=(1,8)          /* Key synchronization
==> field=(1,72),line=80 /* No hex characters
==> maxdiff=5,continue    /* Just in case
==>

```

and hit **ENTER**. After registering them to ISPF, hit **PFK3** to return to the main menu again. Now invoke COMPAREX as a submitted batch job like this:

```
Option ==> 7
```

Now "ISPF Panel CPXJOBST" below will be presented, asking if there are any changes to be made to the JOB card image last used.

```
..... Job Statement .....  
Command ==>  
  
JOB STATEMENT:  
  
==> //youridx JOB (accounting info),MSGCLASS=A, _____  
==> //          CLASS=x,NOTIFY=yourid _____  
==> //* _____  
==> //* _____  
  
Press ENTER to continue;  
Enter END Command to terminate JOB submission.
```

Figure 16. ISPF Panel CPXJOBST

Now overwrite any changes to be made in the JOB card and hit **ENTER**. A message will penetrate the current panel¹ like this:

JOB YOURIDX(JOB02598) SUBMITTED

This is the response to the TSO Submit command. There will be three asterisks "***" at the bottom of the screen. Hit **ENTER** one more time to bring us back to the primary menu. The background job we just submitted should be taking off shortly - depending on how loaded the machine is.

¹ Dependent on how the ISPF interface was installed - TSO Submit or Internal Reader

Index

A

audit trail 13

C

CLIST 16
COBOL
 with TEXT
 example 13
COMPAREX General Flowchart 3
COPYDIFF 4
CPXSISPF 16
CPX@PRIM 17
CPXIFACE 4
CPXJOBST 19
CPXSHORT 17

D

Data File Comparison 5
Data File Comparison - Key and Mask 8
Differences Report 3
differences report 4
Differences Report - DATA
 KEY
 FORMAT
 MASK 9
Differences Report - MAXDIFF Only 7
Differences Report - TEXT=JCL 11
Direct GEM Interface 15
Direct LIBRARIAN Interface 14
Direct LIBRARIAN Interface - Directories 14
Direct PANVALET Interface 13

G

GEM 4
 example 14

I

ISPF Interface 16
ISPF Panel CPX@PRIM 17
ISPF Panel CPXJOBST 19
ISPF Panel CPXSHORT 18
ISPF Panel VENDOR - Raw Form 16
ISPF Panels 16
 CPX@PRIM 17
 CPXJOBST 19
 CPXSHORT 17

VENDOR 16

L

LIBRARIAN 4
 example 14

O

Option == => 17

P

PAN
 with COPYDIFF
 example 13
 with SYSUT1/SYSUT2
 example 13
PAN#1 13
Panels 16
PANVALET 4
 example 13

S

Sample Files - FILE1 And FILE2 4
SYSGEN 16
SYSIN 3
SYSPRINT 4
SYSUT1 4
SYSUT2 4
SYSUT3 4

T

Text File Comparison 10
TSO Command 17

V

VENDOR 16

Y

YES
 with STAMP
 example 13

COMPAREX[®] 6.1.0 (87/054)

User

Reference

OS

Marketing Representative:

STERLING SOFTWARE
Systems Software Marketing Division
11050 White Rock Road
Rancho Cordova, California 95670-6095
Phone: (916) 635-5535
Telex: 377314

Copyright © SERENA Consulting 1987.
Licensed Material.

COMPAREX[®] is a Registered Trademark of:

SERENA Consulting
Post Office Box 117039
Burlingame, California 94010
Phone: (415) 347-0100
Telex: 172040 HQ SMT

Table of Contents

Preface	1
Version, Release, Modification	1
Summary of Amendments	2
ADABAS Enhancements	2
IDMS Enhancements	2
Ignore Sign Differences	2
COPYDIFF=LIB - Expansion of VERS Operand	2
Loosen Restrictions on COPYDIFF=IEBUPDTE	3
DMS - DASD Management System	3
ISPF Interface Rewritten	3
Condor CAMLIB	3
Miscellaneous	3
ASCII File Translation	3
PRINT=FULL with DATA	4
KILLRC Keyword	4
MBRHDR=MATCH	4
CSECT Parsing Address Constants	4
CSECT Parsing BUFFERing	4
Chapter 1 - Major Logical Steps	6
What is COMPAREX	6
What Types of Files Can be Compared	6
General Flowchart and Processing Steps	7
COMPAREX With no Keywords	8
COMPAREX With Keywords	12
Processing of SYSIN	12
Files Opened	12
Reading of Records	12
Comparison of Records - DATA	13
Comparison of Records - TEXT	13
Writing of SYSUT3	13
Writing of SYSPRINT	14
End-of-Job Processing	14
Chapter 2 - Effective Testing	16
Some Effective Testing Flowcharts	16
Checking a Single Program (Unit Testing)	17
Checking a Single Program in a Database Environment	18
Checking a Program Modification (Systems Testing)	19
Checking a System in a Database Environment	20
Managing the Testing Function	21
Set Down Requirements In Writing	21
Develop A Test Plan	21
Garner Approvals For The Test Plan	21
Generating Test Data	22
Functional Department Approval	23
Anticipated Results Are Met	23
Anticipated Results Are Not Met	23
Specifications Are Wrong	23
Chapter 3 - General	24
Legend	24
Sample Execution	25
Keyword General Comments	26

Chapter 4 - Interfaces	28
PANVALET, LIBRARIAN, and GEM	28
PANVALET	29
Potential Error Messages	29
LIBRARIAN	30
Potential Error Messages	30
GEM	30
Potential Error Messages	31
Special Note	31
All DBMS Products	32
ADABAS	33
Potential Error Messages	34
Condor CAMLIB	34
Invoking COMPAREX to Process a CAMLIB	35
Potential Error Messages	36
DL/1	38
Potential Error Messages	39
DMS - DASD Management System	39
Potential Error Messages	39
Disclaimer	40
IAM	40
Potential Error Messages	41
IDMS	42
Parameter Specifications	43
Warning	44
Potential Error Messages	46
ISPF Packed File	46
Disclaimer	47
RAMIS II	47
Potential Error Messages	48
ROSCOE	48
Potential Error Messages	49
WYLBUR	49
Potential Error Messages	50
Roll Your Own	51
Potential Error Messages	52
Synchronizing Databases	52
Comparing	54
ISPF Interface	54
Processing Other Interfaces Disclaimer	54
Connected to Main Menu	55
Sample Sessions	56
DATA Comparison in Foreground	63
TEXT Comparison Via Short Cut	65
Compare Directories Via Edited Batch Job	67
Save and Load Profiles	69
Chapter 5 - Input Processing Keywords	69
Input Processing Keywords	70
SYSIN File Processed	70
Comments	70
HELP	70
Incorrect Keywords	70
Correct Keywords	71
Correct and Incorrect Keywords on Same SYSIN Record	71
End Of Data on SYSIN	71
SYSUT1, SYSUT2, and SYSUT3 Opened	71
SYSUT1 and SYSUT2 Opened	71
SKIPUT1 and SKIPUT2	71
SYSUT3 Opened	71

SYSUT1 and SYSUT2 Read	71
STOPAFT	72
CONTINUE	72
DIRECTORY	72
Displacement	72
Selecting Records for Comparison	72
Keywords	72
Form of Keywords	73
Inclusive Keywords	73
"AND" Logic	73
"OR" Logic	74
Exclusive Keywords	74
"AND" Logic	74
"OR" Logic	74
Pairing Records for Comparison	74
Types of DATA Synchronization	74
TEXT	75
Comparing Only On Specific Fields	75
FIELD	75
MASK	76
IDENTITY	76
Keywords for Input Processing	78
CONTINUE	78
CPXIFACE	78
DATA	79
DESEN	82
DESEN1	82
DESEN2	82
DIRECTORY	83
DIR	83
FIELD	85
FIELD1	85
FIELD2	86
FILTERIN	87
FIN	87
FILTEROUT	88
FOUT	88
FILTORIN	88
FORIN	88
FILTOROUT	89
FOROUT	88
IDENTITY	90
ID	90
IGNORSIN	91
MASK	92
MASK1	92
MASK2	92
MODE	94
SKIPUT1	94
SKIPUT2	94
STOPAFT	95
SYSUT1	96
SYSUT2	97
WILDCARD	98
Chapter 6 - Data File Synchronization Keywords	99
Data File Synchronization Keywords	99
What is DATA?	99
What is DATA Comparison Logic?	99
DUMMY Files	100

KEY Synchronization	100
File Out Of Sequence	100
Duplicate KEY on the Same File	100
KEY Goes Beyond Record	100
End Of Data on SYSUT1	101
End Of Data on SYSUT2	101
Records Matched on KEY	101
Records are Equal	101
Using FIELDS	101
FIELDS Compare Equal	101
Records Compare Unequal	101
No Match on KEY for SYSUT1	101
No Match On KEY for SYSUT2	102
SEGMENT Synchronization	102
Database Out of Sequence	102
Duplicate SEGMENT on Same File	102
SEGMENT Starts Beyond Segment Length	103
End Of Data on SYSUT1	103
End Of Data on SYSUT2	103
Records Matched	103
Records Are Equal	103
Using FIELDS	103
FIELDS Compare Equal	103
Records Compare Unequal	103
Match On SEGMENT, Mismatch On Control Field	103
Ascending Control Field, SYSUT1 Low	104
Descending Control Field, SYSUT1 Low	104
Ascending Control Field, SYSUT2 Low	104
Descending Control Field, SYSUT2 Low	104
No Match On SEGMENT for SYSUT1	104
No Match On SEGMENT for SYSUT2	104
Same-Physical-Record-Number Synchronization	105
End Of Data on SYSUT1	105
End Of Data on SYSUT2	105
Records Compare Equal	105
Using FIELDS	105
FIELDS Compare Equal	105
Records Compare Unequal	105
Keywords Not Available With DATA	106
Advantages of DATA	106
More Efficient	106
Points Out Differing Bytes	106
Disadvantage of DATA	106
Decisions About DATA and TEXT	106
End-Of-Job Counts	107
CPX74I - Bytes Underscored	107
CPX75I - Record Counts	107
CPX76I - Unusable FIELDS, IDENTITYs, SEGMENTs, DESENs	107
CPX78I - Member Counts	107
The DATA Keywords	108
KEY	108
KEY1	108
KEY2	109
SEGMENT	110
SEG	110
Chapter 7 - TEXT Keywords	112
Text Processing Keywords	112
What is TEXT?	112
What is TEXT Comparison Logic?	113

Text Examples	115
Keywords Not Available With TEXT	117
Dummy File	117
KEYs	117
SEGMENTS	117
DATA	117
IDENTITYs	117
FIELDS	117
DESENs	117
MASKs	117
DIRECTORY	118
FLDSONLY	118
LINE	118
NIBBLE	118
PRINTs	118
Advantages Of TEXT	118
Comparing Only On The Significant Data	118
Determining Differences In Context With The Rest Of The File	118
Providing Synchronization When No KEY Is Clear	119
Disadvantages Of TEXT	119
Inefficient	119
Does Not Point Out Differing Bytes	119
Decisions About DATA and TEXT	119
End-of-Job Counts	119
The TEXT Keywords	120
BUFF	120
FRAME	121
MLC	121
SQUEEZE	121
TEXT	122
PRINT	123
Chapter 8 - Output Processing Keywords	124
Output Processing Keywords	124
Writing a File of Selected Records	124
Differing Records Defined	125
SYSUT1 Is A Dummy File	125
KEY Mismatch	125
SEGMENT Mismatch	125
Extra Records On End Of SYSUT2	125
SYSUT2 Record Is Longer Than SYSUT1 Record	125
FIELDS Are Differing	125
With TEXT Comparison	126
No FIELDS	126
With FILTERs	126
End-of-Job Record Count	126
The COPYDIFF Keywords	126
COPYDIFF	126
INSERT	128
DELETE	128
REPLACE	129
SYSUT3	129
Chapter 9 - Display Processing Keywords	130
Display Processing Keywords	130
The All-Defaults Differences Report	131
The All-Defaults Differences Report With DATA	132
Using Keywords To Modify The Differences Report	133
ASCII	133
CASE	133

DASH	134
DECIMAL	134
EBCDIC	135
FLDSONLY	135
FORMAT	136
GENFLDS	138
HALT	138
HELP	139
HEX	141
IGNORSIN	141
INTERLEAVE	142
KEYSONLY	142
KILLRC	142
LINE	142
MAXDIFF	143
MBRHDR	143
NTBBLE	144
PAGE	144
PLUS	146
PRINT	147
Chapter 10 - Messages	148
Messages Issued During Job Initialization	148
CPX00I	148
CPX01I	149
CPX02I	149
CPX03I	150
CPX04I	150
CPX05I	151
CPX06I	151
CPX07I	152
CPX08I	153
CPX09I	154
CPX10I	154
CPX11I	155
CPX12I	155
CPX13I	157
CPX15I	157
CPX16A	159
CPX16I	159
CPX18I	160
CPX19I	160
CPX20I	160
CPX21I	161
CPX22I	161
CPX23I	162
CPX24A	162
CPX25I	163
CPX26I	163
CPX27I	164
CPX30A	164
CPX31A	164
Messages During Processing and End of Job	165
CPX35A	165
CPX36A	165
CPX37A	165
CPX39A	166
CPX40A	166
CPX41I	166
CPX42I	167

CPX51I	168
CPX52I	168
CPX56I	169
CPX57I	169
CPX61I	170
CPX62I	170
CPX64I	170
CPX65I	170
CPX66A	171
CPX67I	171
CPX69I	171
CPX71I	172
CPX72I	172
CPX74I	173
CPX75I	173
CPX76I	174
CPX77I	174
CPX78I	175
CPX80I	175
CPX90A	176
CPX91A	176
CPX92A	177
CPX93A	178
CPX94A	178
CPX97A	178
CPX99A	179
Chapter 11 - Examples	180
Select One Account - FILTERIN	180
Select Two Accounts - FILTERIN's	180
Exclusive Filters	180
Filter Out One Record	181
Filter Out All But Certain Records	181
Filter Out and Filter In	181
Disregard Inserted Records	181
Complex Filtering	181
IDENTITY's and FIELD's	182
COBOL Source Code Changes	182
FILTER's With TEXT	182
Audit PDS Libraries	183
Compare JCL Libraries	183
Regression Test in Database Environment	183
Compare to Backup	183
Find Latest Versions	184
IEBUPDTE Formatting of Audit Trail	184
Desensitize Live Production Data	185
Reverse Delta Deck	185
Appendix A. Sample COBOL Programs	187
COBOL1 - Before Change	187
COBOL1 - After Change	188
Glossary	189
Index	192

List of Illustrations

Figure 1. COMPAREX General Flowchart	8
Figure 2. Sample License Information	9
Figure 3. Installation Defaults	9
Figure 4. Sample Default Messages	9
Figure 5. DATA Example - Modified to Fit Page	11
Figure 6. Messages Issued at End of Job	11
Figure 7. Unit Test in Batch Environment	17
Figure 8. Unit Test in Database Environment	18
Figure 9. Systems Test in Batch Environment	19
Figure 10. Systems Test in Database Environment	20
Figure 11. Sample Batch JCL	25
Figure 12. Sample Foreground CLIST	25
Figure 13. Direct PANVALET Interface	28
Figure 14. Direct PANVALET Interface	29
Figure 15. Direct LIBRARIAN Interface	29
Figure 16. Direct LIBRARIAN Interface	30
Figure 17. Direct GEM Interface	30
Figure 18. ADABAS Interface Record Layout	32
Figure 19. Direct ADABAS Interface	32
Figure 20. Direct CAMLIB Interface	35
Figure 21. DL/1 Interface Segment Layout	36
Figure 22. Direct DL/1 Interface	37
Figure 23. Direct DL/1 Interface	38
Figure 24. Direct DMS Interface	39
Figure 25. Direct IAM Interface	40
Figure 26. IDMS Interface Record Layout	41
Figure 27. Direct IDMS Interface	41
Figure 28. Direct IDMS Interface Via Exit Module - Unload	43
Figure 29. Direct IDMS Interface Via Exit Module - Compare	44
Figure 30. IDMS COBOL Exit Module JOSEFINE - Excerpts	45
Figure 31. Direct ISPF Packed File Interface	46
Figure 32. RAMIS Interface Record Layout	47
Figure 33. Direct RAMIS Interface	47
Figure 34. Direct ROSCOE Interface	48

Figure 35. Direct WYLBUR Interface	49
Figure 36. CPXIFACE Sample Code - Roll Your Own Slot	51
Figure 37. Hierarchical Database Structure - Segment Synchrony	52
Figure 38. Database Structure - Random Key Synchrony	53
Figure 39. ISPF Panel VENDOR - Raw Form	54
Figure 40. ISPF Panel CPX@PRIM	55
Figure 41. ISPF Panel CPXOPDTO	56
Figure 42. ISPF Panel CPXOPDAT	57
Figure 43. ISPF Panel CPXOPSYK	58
Figure 44. ISPF Panel CPXOPDS1	59
Figure 45. ISPF Panel CPXOPDS2	60
Figure 46. ISPF Panel CPXRUNPR	61
Figure 47. ISPF Panel CPXREPRT	62
Figure 48. ISPF Panel CPXSHORT	63
Figure 49. ISPF Panel CPXJOBST	64
Figure 50. ISPF Panel CPXOPDSN	65
Figure 51. ISPF Panel CPXPRSAV	67
Figure 52. ISPF Panel CPXPRLOD	68
Figure 53. Sample Load Modules Composed of CSECTs	80
Figure 54. CSECT Parsing - Execution JCL and Keywords	81
Figure 55. CSECT Parsing - Sample Difference Report	81
Figure 56. IDENTITY, FIELD, MASK, and DESEN Messages	91
Figure 57. Synchronization After FILTERing - No KEY	105
Figure 58. Hierarchical Database Structure	111
Figure 59. TEXT=COBOL	115
Figure 60. TEXT=COBOL and PRINT=FULL	116
Figure 61. FORMAT=02,DECIMAL,NIBBLE (Excerpts)	135
Figure 62. Alphanumeric Format - FORMAT=1y	138
Figure 63. PLUS and FORMAT=21 (Ditto Format)	146
Figure 64. COBOL1 Before Change	187
Figure 65. COBOL1 After Change	188

List of Tables

Table 1. Input Keyword Properties	69
Table 2. Data File Synchronization Properties	99
Table 3. Text Keyword Properties	112
Table 4. Output Keyword Properties	124
Table 5. Display Keyword Properties	130

Preface

COMPAREX runs under the three major IBM operating systems:

- OS/VS - (*MVS, MVS/XA*)
- DOS/VSE - (*SP 2.1 System Package, and pre-2.1*)
- VM/CMS

In addition, it runs under these operating systems:

- MVT/VSE from Software Pursuits
- OSIV/F4 from FUJITSU/FACOM

Version, Release, Modification

All software created and maintained by SERENA Consulting will have a Version, Release, and Modification level associated with it. Only when the Version or Release number changes (*usually annually*) will the full customer base (*those who are up to date with maintenance*) be issued new tapes and documentation.

There will be at least two interim Modifications during the year. At year end, the accumulated Modifications will be rolled into the next general Release.

This manual describes:

COMPAREX OS
Version 6
Release 1
Modification 0
Julian release date 87/054

This COMPAREX 6.1.0 **User Reference Manual** gives an extensive discussion of the functions and processing logic of COMPAREX, the comparison utility. The User Reference Manual has been written for the professional programmer who has experience with programming, file structures and organizations, utilities, and testing practices. The manual contains extensive information about the COMPAREX keywords and messages.

The COMPAREX 6.1.0 **Quick Start Manual** is a reference document that gives the information most needed by COMPAREX users. It illustrates, by example, how to invoke COMPAREX for solving some of the most common comparison situations.

Finally, the COMPAREX 6.1.0 **User Reference Card** shows the format and the description of each keyword.

Summary of Amendments

COMPAREX 6.1.0 has the following enhancements over the prior (6.0) release as of February 23, 1987.

ADABAS Enhancements

In addition to the *Read Physical* - L2 call, the following functions are also available:

- Specify the Database and File
- Read logically by a particular key field - L3 call
- Specify fields to be retrieved - format buffer
- Process Natural programs - source and object - directly from their ADABAS database and file

See "ADABAS" on page 32.

IDMS Enhancements

In addition to the *Area Sweep within record type*, a new facility is also available. There are now two major ways to process the database. The first is a slight variation on the area sweep. The second is to specify an exit module (*written in your own programming language*) to navigate through the database.

See "IDMS" on page 41.

Ignore Sign Differences

When comparing DATA files and the basic difference between them is that one file contains packed fields with a sign of C and the other contains packed fields with a sign of E, every record contains differences. That can be overcome with a new keyword:

IGNORSIN

See "IGNORSIN" on page 141.

COPYDIFF=LIB - Expansion of VERS Operand

When creating a Delta Deck in LIBRARIAN format, the VERS operand may be used to create or suppress the version on the Select card image. The only options were YES or NO. Now there is a third option called

YESHMM

which appends the hour and minute to the month and day such that a single module within multiple ones created on the same day can be isolated by the LIBRARIAN.

See "COPYDIFF" on page 126.

Loosen Restrictions on COPYDIFF=IEBUPDTE

When comparing two partitioned datasets (PDS) and creating a Delta Deck in IEBUPDTE format, the sequence numbers may be taken from any columns with relatively few restrictions.

See "COPYDIFF" on page 126.

DMS - DASD Management System

This interface is intended to be a "starter set" of a more complex interface to archived datasets. In this release, we have concentrated on accessing large archived VSAM datasets.

See "DMS - DASD Management System" on page 39.

ISPF Interface Rewritten

The ISPF Interface has been rewritten such that it is now mainly Panel Driven as opposed to Program Driven. See "ISPF Panel VENDOR - Raw Form" on page 54 for the new method of invoking the interface. This implies that the jump function between panels is now allowed.

See "ISPF Interface" on page 54.

Condor CAMLIB

This read-only interface processes the library structure known as CONDOR CAMLIB. You may compare the directories, individual members or the entire library to a PANVALET, LIBRARIAN, or another CAMLIB library.

See "Condor CAMLIB" on page 34.

Miscellaneous

The following miscellaneous enhancements have been made.

ASCII File Translation

The source code to CPXIFACE now has a new global variable

```
&ZASCII  SETB  1
```

which can be turned on at will at "gen" time for the interface to have the mechanics of translating ASCII bytes to EBCDIC bytes for subsequent comparison. It generally is used within the "Roll Your Own" slot:

```
&ZROLOWN SETB  1
```

In actual execution, it translates each ASCII byte into an EBCDIC byte just before passing the entire record 'up' to COMPAREX. It can be invoked, with minimal programming effort, by moving the invoking instruction

```
BAL  LINKREG1,ASCII2EBC
```

to the necessary spot after reading the individual record.

It does a wholesale translation, regardless of content. For this reason, it is recommended that files containing ASCII Packed numeric data be given some extra programming attention in the interface because they translate literally not numerically.

PRINT=FULL with DATA

In the past, when the user specified (or defaulted to) 'DATA', and there were differences in some records, only the differing records were displayed. Within 'TEXT' logic, there is an option to the PRINT keyword,

PRINT=FULL

that would force the printing of all records with the differences interspersed. Now that option has meaning in DATA logic as well.

See "PRINT" on page 147.

KILLRC Keyword

This is a very rarely used keyword. It forces a return code of zero to be passed back regardless of what the outcome of the compare was. It was created specifically to keep DOS 2.1 SP compatible with OS and VM/CMS. The only users who have to use it are DOS pre-2.1 releases such as 1.3.5 and they are in the distinct minority.

See "KILLRC" on page 142.

MBRHDR= MATCH

The current options to keyword **MBRHDR** are

- YES
- NO
- COND

A fourth option is now available called

MATCH

which is similar to **CONDitional** except that member names that do not match are not reported.

See "MBRHDR" on page 143.

CSECT Parsing Address Constants

If one Csect changed slightly, it sometimes happens that the V-type address (ADCON) constants are all that change in the rest of the Csects. You may or may not be interested in seeing the differences in the address constants. Now there is an alternative. You may specify:

DATA= (CSECT , ADCON=NO)

to have each address constant (A-type and V-type) nullified to binary zeroes before any comparison is done. The default is **ADCON= YES**, implying that address constants are to be left alone. **ADCON=NO** should result in a shorter difference report.

See "DATA" on page 79.

CSECT Parsing BUFFering

When parsing CSECT's, the **BUFF** parameter defaults to 256. It is also explicitly recommended that **BUFF** be set higher such as:

BUFF=1000

Now the reason for setting a higher BUFFER is even greater. If the buffer cannot fully contain both load modules (at buffer stocking time) then the comparison is terminated with message:

CPX39A - DATA RECORDS TOO LARGE FOR BUFFER - (xxxxxxx,nnn) - FUNCTION TERMINATED - RETURN CODE = 16

Chapter 1 - Major Logical Steps

COMPAREX is a computer program used to compare two magnetically recorded files and to print a report showing the records that are different.

Data processing professionals in charge of the accuracy of systems implementations and modifications use COMPAREX to help to ensure the correctness and completeness of their work.

When a change is introduced into a system, COMPAREX compares the files produced by the system prior to the change with the files produced by the system after the change. The Differences Report shows the records that are not exactly alike, identifying the file and the actual record number on the file for each record printed.

During an examination of the Differences Report, the user can see exactly which bytes in the two records are differing. The user looks for the expected changes and, more importantly, the unexpected changes.

With the use of extensive optional keywords, the user can modify COMPAREX's input processing, record pairing, and printing routines. The most frequently requested options are implemented as the COMPAREX defaults, and the first-time user can run an effective COMPAREX job using no keywords.

COMPAREX is **THE** comparison utility. It helps the conscientious user check the accuracy of maintenance changes before implementation, and it facilitates effective unit and system testing for new development.

What is COMPAREX

COMPAREX is a utility designed to compare two files and to write a report that displays the records that are different.

COMPAREX can be run in an all-defaults mode; it reads two files comparing each record from the first file with the same numbered record on the second file, and it prints both records only when it finds that at least one byte is different. In this all-defaults mode, COMPAREX reads to the end of each file, printing all pairs of records with differences and printing all extra records from the longer input file.

The all-defaults mode can be significantly modified by the use of COMPAREX's keywords. In a keywords job, the user specifies free-form keywords to change the default parameters. In this way, a file of any organization can be read, synchronization can be done by logical keys, and the format of the report can be customized. In addition, keywords can be used to tell COMPAREX to create an output file of selected records.

What Types of Files Can be Compared

COMPAREX can compare any two magnetically recorded files of almost any structure or organization. A fairly exhaustive list of the types of files COMPAREX can directly process follows:

- Program
 - Source Code
 - Object Code
 - Load Modules
 - ▲ Csect Parsing or Undefined Block to Block
- Job Control Language (JCL)
- CLIST

- System Master Files
 - QSAM
 - ISAM
 - VSAM
- System Intermediate Files
- System Transaction Files
- Directories, Selected Members, Ranges of Members or All Members
 - PANSOPHIC'S PANVALET
 - Applied Data Research's LIBRARIAN
 - FUJITSU/FACOM'S GEM
 - Partitioned DataSet (PDS)
- Database for Most Database Management Systems (DBMS)
 - ADABAS
 - DL/1 (IMS, IMS/FASTPATH, CICS)
 - IAM
 - IDMS (5.7) or IDMS/R (10.0)
 - RAMIS II
 - ROSCOE
 - WYLBUR
 - Others to be developed in the future.
- Unformatted Screens (PANELS) for Online Terminals
- Control Card Images
- Reports
- Documentation (such as raw SCRIPT/VS input)
- ISPF 2.1 Packed Files (PACK ON)

General Flowchart and Processing Steps

"COMPAREX General Flowchart" on page 8 shows the five COMPAREX files.

INPUT - three files are used:

- SYSIN contains any keywords to modify the all-defaults mode. If SYSIN is not able to be opened or if SYSIN is empty, COMPAREX processes without any keywords, taking all defaults. If SYSIN is not empty, COMPAREX modifies its default processing with the user's keywords.
- SYSUT1 is the first file of the two to be compared. Usually, this is the **Baseline** or Old file. If program source code is being compared, this is the unmodified version.
- SYSUT2 is the second file of the two to be compared. Usually, this is the **Testline** or New file. If program source code is being compared, this is the updated version.
- If PANVALET, LIBRARIAN, GEM, or OTHER file structures are to be processed directly, the proper DDNAME must be supplied instead of SYSUT1 or SYSUT2 or both so that the COMPAREX interface (CPXIFACE) can access it.

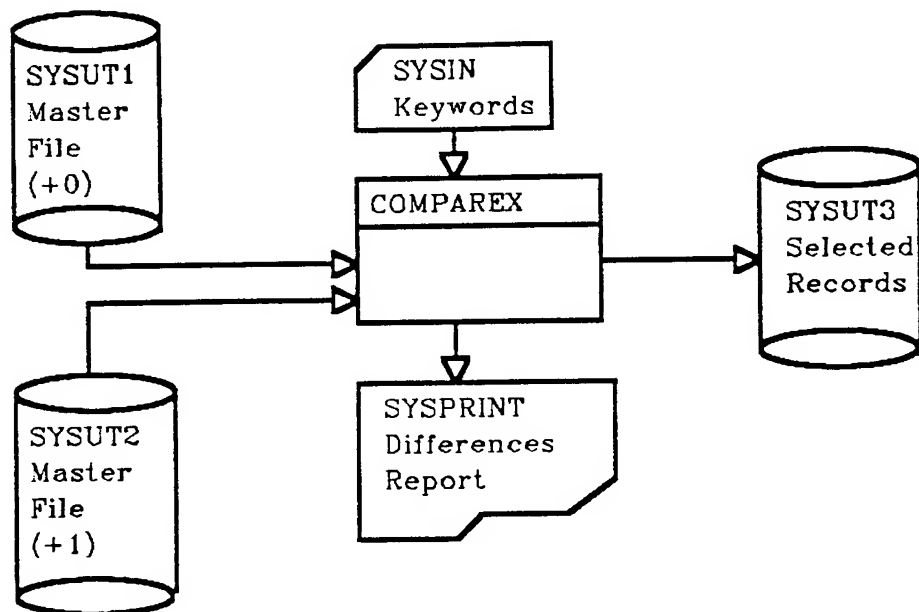


Figure 1. COMPAREX General Flowchart

OUTPUT - two files are generated:

- **SYSPRINT** contains the Differences Report, a listing of the results of the comparison. It also contains a list of the defaults and keywords used, and it shows end-of-job statistics lines. The user can also specify a list of the COMPAREX keywords and a visual representation of each record type identified.
- **SYSUT3** is an optional file. If present, it contains differing records from file SYSUT2. COMPAREX can be used as a test file generator by the specification of COPYDIFF (the keyword that directs COMPAREX to write file SYSUT3).

COMPAREX With no Keywords

If no keywords are specified, this is the order of the utility's processing:

- **SYSPRINT** is opened. If the open is not successful, COMPAREX abends with a user abend code of 5.
- COMPAREX displays the license information on SYSPRINT. See "Sample License Information" below.

COMPAREX (OS-6.1.0 - 87/054)

MONDAY FEBRUARY 23, 1987 (87/054) 17:11:59 PAGE 1

PROPRIETARY SOFTWARE PRODUCT OF SERENA CONSULTING PHONE (415)347-0100 OR TELEX 172040 HQ SMT

LICENSED TO: your corporate name/trial offer expires ...
your city, state, zip/agent contact for license ...

ALL OTHER RIGHTS RESERVED - USE OF THIS SOFTWARE
 PRODUCT BY UNAUTHORIZED PERSONS IS PROHIBITED.

Figure 2. Sample License Information

- As delivered, COMPAREX has some installation defaults which can override the normal defaults as described in this manual. Your installation may have changed them to fit your own shop's needs. Refer to "Installation Defaults" on page 9 for the delivered installation defaults.

```
CPX00I - *****
CPX00I - ***  I N S T A L L A T I O N  D E F A U L T S  ***
CPX00I - *****
CPX00I -      HALT=COND      /* STOP EXECUTION IF SYNTAX ERRORS FOUND */
CPX00I - *****
CPX00I - ***  END OF INSTALLATION DEFAULTS  ***
CPX00I - *****
```

Figure 3. Installation Defaults

- COMPAREX attempts to open SYSIN. If this open is not successful, COMPAREX issues warning message CPX02A. If SYSIN is able to be opened but is empty (since no COMPAREX keywords are being specified for this no-keywords job), COMPAREX continues to process without issuing the warning message.
- COMPAREX displays on SYSPRINT the defaults it will use. "Sample Default Messages" on page 9 shows a sample of the default messages.

```
CPX03I - EXECUTION OF TS972.CPX.SC1551 - VALUES EXTRACTED/DEFAULTED:
CPX04I - MAXDIFF=999999999999,STOFAFT=999999999999
CPX05I - PRINT=(MATCH,MISMATCH),MBRHDR=YES,HALT=COND
CPX06I - WILDCARD=C'.' ,MODE=APPLICATIONS,(ALL DISPLACEMENTS RELATIVE TO ONE)
CPX08I - DECIMAL,EBCDIC,CASE=UPPER,LINE=(32,HORIZONTAL),PAGE=58
CPX11I - DASH=C'-' ,PLUS=C'+'
CPX21I - SYSUT1=SERE001.X1.SEGMENTS          DCB=(DSORG=PS,RECFM=F,BLKSIZE=80)
CPX22I - SYSUT2=SERE001.X2.SEGMENTS          DCB=(DSORG=PS,RECFM=F,BLKSIZE=80)
CPX25I - DATA,FORMAT=02,INTERLEAVE=0
        (FORMAT EXPLANATION: FULL SYSUT1 FOLLOWED BY DIFFERING LINES OF SYSUT2)
```

Figure 4. Sample Default Messages

Here is a complete list of the defaults COMPAREX takes:

- Input Processing
 - CONTINUE is not in effect
 - DATA file comparison logic is in effect
 - DIRECTORY's are not processed
 - FIELD's are not used
 - FILTER's are not used
 - IDENTITY's are not used
 - MASK's are not used

- MODE=APPLICATIONS is in effect
- SKIPUT1 - no records are bypassed on SYSUT1
- SKIPUT2 - no records are bypassed on SYSUT2
- STOPAFT=999999999999 is in effect
- SYSUT1 is specified by JCL
- SYSUT2 is specified by JCL
- WILDCARD is not used
- Data Files Synchronization
 - KEY's are not used
 - SEGMENT's are not used
- Text File Processing
 - Under the no-keywords job, text file processing is not done; text file keywords (BUFF, FRAME, MLC, PRINT=FULL, SQUEEZE, and TEXT) are not used.
 - Under the no-keywords job, data file comparison logic is in effect. DATA and TEXT comparison logic are mutually exclusive, so TEXT comparison logic is not in effect.
- Output Processing
 - Under the no-keywords job, output processing is not done; output processing keywords (COPYDIFF, and SYSUT3) are not used.
- Display Processing
 - ASCII is not in effect; EBCDIC is in effect
 - CASE=UPPER is in effect
 - DASH=C'-' is in effect
 - DECIMAL is in effect
 - EBCDIC is in effect
 - FLDSONLY is not in effect
 - FORMAT=02 is in effect
 - No GENFLDS are generated
 - HALT=NO is in effect
 - No HELP listing is produced
 - HEX is not in effect; DECIMAL is in effect
 - INTERLEAVE=0 is in effect
 - LINE=(32,HORIZONTAL) is in effect
 - MAXDIFF=999999999999 is in effect
 - MBRHDR=YES is in effect
 - NIBBLE is not in effect
 - PAGE=58 is in effect
 - PLUS=C'+' is in effect
 - PRINT=MATCH and PRINT=MISMATCH are in effect
- Unless HALT=YES has been specified, COMPAREX opens SYSUT1. If the open is not successful, COMPAREX terminates, issuing message CPX90A and a return code of 16.

- Unless HALT=YES has been specified, COMPAREX opens SYSUT2. If the open is not successful, COMPAREX issues a warning with message CPX90A, and unless HALT=COND has been specified, COMPAREX functions as a print utility, printing all records of SYSUT1 onto SYSPRINT.
- COMPAREX reads SYSUT1 and SYSUT2 sequentially. It compares each SYSUT1 record with the same numbered record on SYSUT2 (that is, record number 1 on SYSUT1 is compared with record number 1 on SYSUT2 and record number 1001 on SYSUT1 is compared with record number 1001 on SYSUT2).
- COMPAREX, when it finds that compared records do not match exactly (all bits in all bytes are not equal), prints both records on SYSPRINT and underscores the differences. "DATA Example - Modified to Fit Page" on page 11 shows how two differing records might look on the Differences Report.

CPX61I - KEY SYNCHRONIZATION MISMATCH - RECORD 1 ON FILE SYSUT1

0000	F0F0F0F4	F5F6F7F9	00000000	0045679C	0000B26F	C4C1E3C1	...	*00045679.....?DATA-SYSUT1 *	O N E
0020	F0F1F2F3	F4F50000	00000012	345C0000	30394040	40E3C8C9	...	*012345.....*.... THIS IS THE*	O N E
0040	40C4C1E3	C140C6D6	D940E2E8	E2E4E3F1	40606060	60606040	...	* DATA FOR SYSUT1 ----- *	O N E
0060	40404040	40404040	40404040	40404040	40404040	40404040	...	* * * * *	O N E

CPX51I - RECORD NUMBER 2 ON FILE SYSUT1
CPX52I - RECORD NUMBER 1 ON FILE SYSUT2

0000	F0F0F0F4	F5F6F8F8	00000000	0045688C	0000B278	C4C1E3C1	...	*00045688.....DATA-SYSUT1 *	O N E
0000	F0F0F0F4	F5F6F8F8	00000000	0045688C	EA00B278	C4C1E3C1	...	*00045688.....DATA-SYSUT2 *	T W O
					..				-DIFFERENCE+
0020	F0F1F2F3	F4F50000	00000012	345C0000	30394040	40E3C8C9	...	*012345.....*.... THIS IS THE*	O N E
0040	40C4C1E3	C140C6D6	D940E2E8	E2E4E3F1	40606060	60606040	...	* DATA FOR SYSUT1 ----- *	O N E
0040	40C4C1E3	C140C6D6	D940E2E8	E2E4E3F2	40606060	60606040	...	* DATA FOR SYSUT2 ----- *	T W O
									-DIFFERENCE+
0060	40404040	40404040	40404040	40404040	40404040	40404040	...	* * * * *	O N E
0060	40404040	40404040	40404040	40404040	40404040	40404040	...	* * * * *	T W O
								++++++ *	-DIFFERENCE+
0080	4040							* * * * *	T W O
	++++							++	-DIFFERENCE+

Figure 5. DATA Example - Modified to Fit Page

- If SYSUT1 has come to end of file and SYSUT2 has not come to end of file, COMPAREX prints all SYSUT2's records on SYSPRINT until SYSUT2's end of file has been reached.
- Likewise, if SYSUT2 has come to end of file and SYSUT1 has not come to end of file, COMPAREX prints all SYSUT1's records on SYSPRINT until SYSUT1's end of file has been reached.
- COMPAREX issues its statistics on SYSPRINT. "Messages Issued at End of Job" on page 11 shows sample end-of-job messages.

CPX67I - MAXDIFF INVOKED, CONTINUING WITHOUT PRINTING BY REQUEST

CPX71I - END OF DATA ON FILE SYSUT1

CPX72I - END OF DATA ON FILE SYSUT2

CPX74I - BYTES UNDERSCORED(15)

CPX75I - RECORDS PROCESSED: SYSUT1(28)/SYSUT2(34),DIFFERENCES(3,1,7)

EXPLANATION - 3 RECORDS DIFFER THAT SYNCHRONIZED TOGETHER

1 RECORD WAS CONSIDERED INSERTED ON SYSUT1

7 RECORDS WERE CONSIDERED INSERTED ON SYSUT2

CPX80I - TIME OF DAY AT END OF JOB: 23:57:02 - CONDITION CODE ON EXIT: 4

Figure 6. Messages Issued at End of Job

- COMPAREX closes all files.

COMPAREX With Keywords

COMPAREX accepts a complete set of keywords to modify the no-keywords processing. When COMPAREX is able to open SYSIN and the utility finds one or more valid keywords in SYSIN, this is the order of the utility's processing:

Processing of SYSIN

SYSPRINT file is opened and the license information is displayed on SYSPRINT, exactly as described in COMPAREX WITH NO KEYWORDS.

COMPAREX opens SYSIN, and the utility looks at each SYSIN record to find its keywords. Each SYSIN record is listed on SYSPRINT, to the right of message CPX00I. If the first character of the SYSIN record is an asterisk, COMPAREX considers the entire SYSIN record to be a comment, and the utility does not search for keywords on that record. Additional comments may be placed to the right of keywords by preceding them with a slash-asterisk "/*". If COMPAREX finds information on a SYSIN record that cannot be recognized as a valid keyword, the utility underscores the characters and displays the literal "ERROR?" in the right-hand column.

If the HELP keyword was found in the SYSIN file, COMPAREX displays the CPX01I message and a listing of each valid keyword, its correct format, and a short description of its use.

COMPAREX uses the valid keywords from the SYSIN file to modify the defaults, and the utility displays messages CPX03I through CPX12I to show these processing parameters.

If GENFLDS was specified, COMPAREX creates a visual representation of the fields to be compared for each record type as specified by the IDENTITY keywords.

Files Opened

COMPAREX uses the SYSUT1, SYSUT2, and SYSUT3 keywords to determine the dataset organization and dataset attributes for DOS users. CMS users tell COMPAREX the file name, file type, and file mode. Then COMPAREX interrogates the system for file attributes. Dataset organization and dataset attributes under OS are taken from the operating system but may be overridden by the user.

If COPYDIFF was specified, COMPAREX opens file SYSUT3. Message CPX16I shows the dataset organization and attributes.

If SYSUT1=DUMMY was not specified and HALT=YES was not specified, COMPAREX opens file SYSUT1. Message CPX21I shows the dataset organization and attributes.

If SYSUT2=DUMMY was not specified and HALT=YES was not specified, COMPAREX opens file SYSUT2. Message CPX22I shows the dataset organization and attributes.

If TEXT processing is specified and COMPAREX has not nullified TEXT because of other keywords, the utility issues message CPX25I, showing the TEXT file keywords in effect for the run.

If TEXT processing is not being done for the run, COMPAREX issues message CPX25I, showing that DATA file synchronization is in effect.

If either SKIPUT1 or SKIPUT2 was specified, COMPAREX skips over the specified number of records on the input.

Reading of Records

If the STOPAFT keyword has been specified, COMPAREX reads records until that keyword's number has been reached. Otherwise, COMPAREX reads records until it has read 99999999999 records on either file.

If MAXDIFF has been specified, and COMPAREX has already printed the maximum number of differences specified with the MAXDIFF keyword on the Differences Report, the utility continues to read records if CONTINUE has been specified or it goes to its end-of-processing routines if CONTINUE has not been specified.

COMPAREX considers all record displacements on keywords to be relative to one unless the MODE=SYSTEMS keyword has been entered to change displacements to be relative to zero.

COMPAREX reads records and pairs them for comparison. If TEXT comparison logic is in effect, COMPAREX uses buffers and look-ahead logic to pair records and to isolate differing blocks. Under TEXT processing, keywords may be entered to control the size of the look-ahead logic buffer (BUFF), the back-in-synchronization matching line count (MLC), the characters to be deleted (SQUEEZE), and the format of the Differences Report (FRAME and PRINT=FULL). If DATA comparison logic is in effect, COMPAREX pairs records based on KEYs, SEGMENTs, or same physical-record-number synchronization.

COMPAREX uses any filtering keywords, along with the WILDCARD value, to determine which records are sent to the comparison routines.

Comparison of Records - DATA

At the point where COMPAREX compares records, the default processing (essentially, the comparison of all bytes) can be changed by keywords.

The comparison of DATA is described here. DATA is used when files have an inter-record relationship. TEXT file comparison is also done by COMPAREX, and is described on the next page. DATA is the default.

If a record has been sent to the comparison routines alone, identified as a key synchronization mismatch, a segmenting synchronization mismatch, or an extra record, the utility sends this record to the Differences Report without doing any comparison.

If no IDENTITY, FIELD, or MASK keywords have been specified, COMPAREX compares all bytes of the two records. If any one byte is different or if one record is longer than the other, COMPAREX identifies this pair as different and sends the pair of records to the Differences Report. If all bytes in the two records are equal, COMPAREX does not send this pair to the Differences Report.

If IDENTITY, FIELD, or MASK keywords have been specified, COMPAREX uses these keywords to compare the various parts of the record. IDENTITY keywords test for a value on the SYSUT1 record so that the following FIELDs and MASKs can apply to that record type only. FIELD keywords specify bytes to be compared, and MASK keywords specify bytes that are to be ignored. If any bytes in FIELDs are unequal, COMPAREX identifies this pair as different and sends them to the Differences Report, noting which IDENTITY and FIELD uncovered the first difference.

Comparison of Records - TEXT

If Text comparison has been specified, COMPAREX changes its comparison routines to try to match up records by the values in the records. Text comparison is used for program source code, JCL, and documentation.

No KEY or SEGMENT can be specified for synchronization. COMPAREX synchronizes records by attempting to find records where all positions of the record are equal. Inserted records are identified as being those between matched records.

Any character can be removed from the record prior to the comparison with the SQUEEZE keyword. The BUFF keyword directs the storage size for buffering, and the MLC keyword tells COMPAREX how many equal compares to make before identifying a back-in-synchronization condition. The PRINT=FULL keyword can be used with Text to print all records on SYSUT1, whether they are unmatched or not, and the FRAME keyword is used to surround blocks of records on the Differences Report.

Under Text comparison, both input files must be present. IDENTITYs, FIELDs, and MASKs are not used.

If TEXT comparison is specified, COMPAREX identifies only differing records on the Differences Report. Differing bytes are not underscored.

Writing of SYSUT3

If COPYDIFF was specified, and if SYSUT3 was successfully opened, the utility writes any record from file SYSUT2 that it identified as differing onto file SYSUT3. These differing records include matched records where some difference in data

is found and records that are Inserted on SYSUT2.

Note: Records inserted on SYSUT1 do not go to SYSUT3.

Writing of SYSPRINT

The Differences Report shows the differing records. The format of the Differences Report is modified by the display processing keywords specified. The input file and the associated logical record number on that input file are shown with each record printed, and the message on the Differences Report also shows why the record was identified as differing.

Many keywords are available to direct COMPAREX to modify the default parameters for the printing of the Differences Report.

The HELP keyword causes COMPAREX to print a listing of valid keywords and their descriptions on the Differences Report.

The PRINT keyword directs COMPAREX to print synchronized matched records or mismatched records.

COMPAREX translates the input to characters for printing using its EBCDIC translate table. The user may specify that COMPAREX use an ASCII translate table instead.

The MAXDIFF keyword specifies the maximum number of differing records, mismatched records, and extra records to print on the Differences Report. A MAXDIFF keyword should be included in every COMPAREX job to avoid large printouts when the expected results do not occur.

COMPAREX shows each line's relative displacement in decimal. The user may specify that COMPAREX show the relative displacement in hexadecimal instead, by the use of the HEX keyword.

The GENFLDS statement causes COMPAREX to print out a handy visual representation of each record type, as identified by IDENTITIES, FIELDS, and MASKS. These sheets can be made into clear plastic overlays on a copying machine and used in a review of the Differences Report.

The LINE keyword changes the number of bytes shown on each line of the Differences Report, the PAGE keyword changes the number of print lines on each page, and the FORMAT keyword specifies formatting characteristics on how differences are displayed and permits INTERLEAVING of displayed lines.

COMPAREX underscores each differing byte with a dash. The user can change this to any other character by the use of the DASH keyword.

COMPAREX underscores excess bytes on the record from file SYSUT2 with a plus. The user can change this to any other character by the use of the PLUS keyword.

In general, COMPAREX underscores both half-bytes¹ of each differing byte. The user can change the Differences Report to see only differing half-bytes underscored by the use of the NIBBLE keyword. COMPAREX will underscore all differing bytes, even those bytes considered MASK'ed out unless FLDSONLY is also specified.

End-of-Job Processing

If the input files contain an embedded directory such as a partitioned dataset (PDS), PANVALET master or LIBRARY master, COMPAREX issues an end-of-data or end-of-text message at the end of each member - unless DIRECTORY processing was requested. Also, with directory-embedded files, an end-of-directory message is issued when each directory is exhausted. If the input files are not directory-embedded, COMPAREX issues an end-of-data or end-of-text message at the end of each input file.

When keywords have been used, COMPAREX modifies the default processing to show additional counters.

The statistics line, message CPX75I, shows the number of records written onto SYSUT3, if COPYDIFF was specified.

If KEY or SEGMENT synchronization is used, COMPAREX shows, next to 'DIFFERENCES,' as the left-hand figure, the number of pairs of records which were synchronized by KEY or SEGMENT and some difference was found; as the

¹ Only in the Dump format - FORMAT=0y - horizontal hex

middle figure, the number of records from file SYSUT1 that were not synchronized to any SYSUT2 record; and, as the right-hand figure, the number of records from file SYSUT2 that were not synchronized to any SYSUT1 record.

If **FIELDs**, **SEGMENTs**, **IDENTITYs**, or **DESENs** are specified, COMPAREX will show, with message CPX76I, the number of these that went beyond the length of the record.

If filtering keywords are used, COMPAREX will show, with message CPX77I, the number of records that were not passed to the comparison routines due to filtering tests.

The clock time at the end of the job is shown, and the condition (return) code is displayed (0, 4, and 8 indicate normal processing; condition code 16 indicates a serious error, and this error is always described by another message on **SYSPRINT**, printed to the right of a message number ending with 'A' for ACTION).

COMPAREX closes all files.

Chapter 2 - Effective Testing

The most important step in the testing of program and system changes is to compare the actual results with the anticipated results and to reconcile the differences.

This comparison can be done manually or with COMPAREX, the comparison utility. The manual comparison process is both time-consuming and subject to error; each byte of each record produced should be compared with what was expected. Because the manual process is so time-consuming, often the tester checks only the fields of greatest concern and forgets to examine necessary literals and keys.

COMPAREX compares every byte unless specifically instructed to do otherwise. Forgotten literals and keys present themselves boldly on the Differences Report, and the tester can correct the code before it is put into production.

COMPAREX should not be used sparingly. On the first execution during a testing session, the user should specify few keywords and allow COMPAREX to take its defaults. On this first execution, MAXDIFF=10 could be specified to limit the differences shown and CONTINUE could be specified to ensure that all records are read and produce statistics if needed.

The user reviews this first Differences Report to see how many inequalities were found. The end-of-processing messages show the total number of differing pairs of records. Then, the user runs COMPAREX again, using KEYS, FIELDS, MASKS, and FILTERS to properly synchronize the files and to correctly select records and data within those records for comparison.

In this way, each COMPAREX run reveals more about the differences between the two files and, at the same time, more about the differences between the two programs that created them. Errors in programs are first discovered by an examination of the data they produce, and additional COMPAREX jobs can be run, using the TEXT keyword, to compare two versions of source code to identify added, changed, and deleted source lines.

This chapter on effective testing presents information on using COMPAREX to check the correctness of a single program or an entire system. In addition, the user interested in the management of testing procedures will find a discussion of test plans and test data at the end of this chapter.

Some Effective Testing Flowcharts

"Unit Test in Batch Environment" on page 17 through "Systems Test in Database Environment" on page 20 show some ways to effectively use COMPAREX in testing.

Checking a Single Program (Unit Testing)

Measuring the effect of transactions on a master file update program can often prove the effectiveness of a program. "Unit Test in Batch Environment" below shows a flowchart for such a measurement.

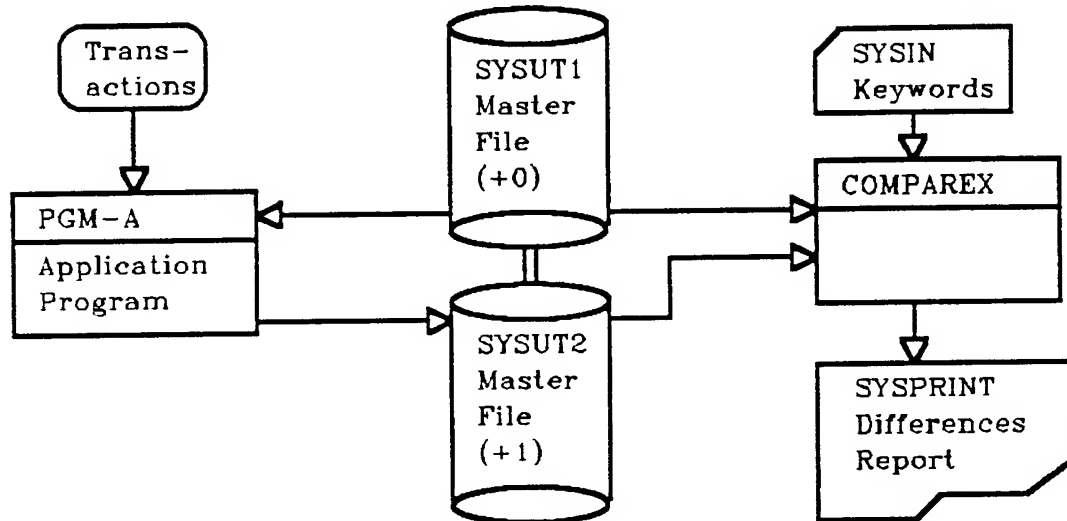


Figure 7. Unit Test in Batch Environment

The master file from the previous update is used as input to the master file update, along with a file of transactions.

The previous master file and the master file created by the test run are used as input to COMPAREX. In this COMPAREX run, the old master file is used as SYSUT1 and the new master file (*the +1 version as shown on "Unit Test in Batch Environment"* is used as SYSUT2).

Also used as input to COMPAREX is a file of specifications containing the COMPAREX keywords. When master files are being compared, KEY synchronization usually is specified. This allows COMPAREX to identify inserted and deleted records. Any date/time stamp on the two files could be ignored in the comparison by a MASK keyword.

The Differences Report is the proof of the effectiveness of the program. Each inserted and deleted record is shown, and the user reviews these to reconcile them to the anticipated results. Any changed record is also reviewed. The differing bytes of the record are reconciled to the anticipated results.

After the Differences Report has been received, the user either is satisfied with the program's effectiveness or has a list of deficiencies to be corrected. After any program corrections, the user runs COMPAREX until all differences have been reconciled.

Checking a Single Program in a Database Environment

Measuring the effect of a change to a program updating a database through a database management system such as IDMS is similar to the checking of a single program, as discussed above. The difference is that the database may be processed directly or unloaded to a Flat File before processing. There are advantages and disadvantages to each method. "Unit Test in Database Environment" below shows a flowchart for such a procedure.

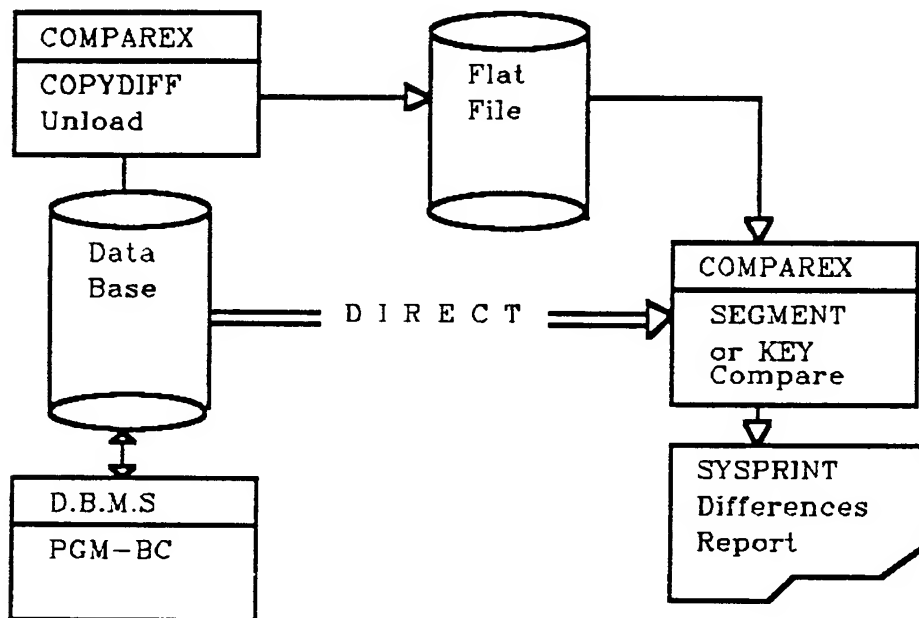


Figure 8. Unit Test in Database Environment

Prior to the execution of the new code, the database is unloaded to a Flat File with COMPAREX. Then, the new code is executed against the database. Finally, the Flat File is compared against the just updated database. Variations on the SEGMENT keyword or KEY keyword are used to synchronize between the two databases.

Again, the Differences Report is the proof of the effectiveness of the program modification. The user reviews the report to reconcile the actual results to the anticipated results. The user makes necessary program corrections, restores the database, and reruns the program and COMPAREX until all differences have been reconciled.

Checking a Program Modification (Systems Testing)

"Systems Test in Batch Environment" on page 19 shows a flowchart for measuring the effect of a program modification.

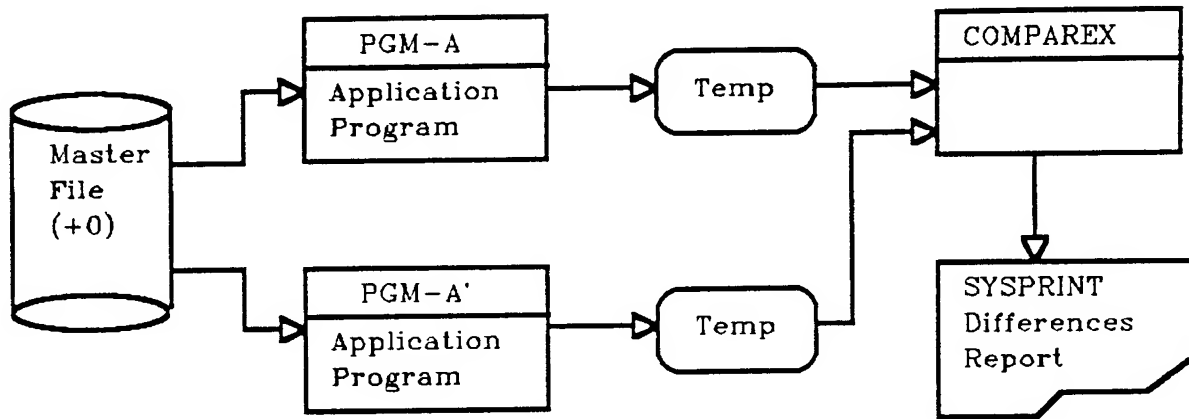


Figure 9. Systems Test in Batch Environment

The old program is run, using the current master file as input; and the modified program is run, using this same current master file as input. Then, COMPAREX is used to compare the output TEMP and TEMP' files.

The versions of the files created by the unmodified program (the baseline files) are used as SYSUT1 in the COMPAREX runs, and the versions of the files created by the modified program (the testline files) are used as SYSUT2 in the COMPAREX runs.

The two Differences Reports are reviewed to evaluate the effectiveness of the program modification. The user corrects any deficiencies in the testline program and runs the series of programs again.

Checking a System in a Database Environment

"Systems Test in Database Environment" below shows a flowchart for measuring the effectiveness of a program change in a complex database environment.

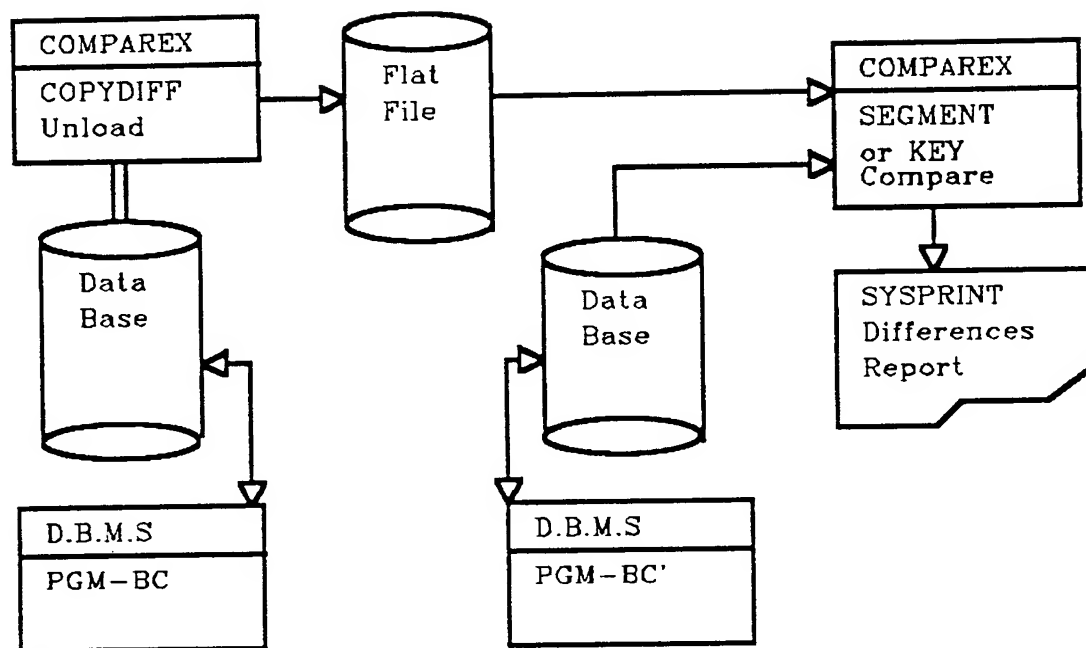


Figure 10. Systems Test in Database Environment

Systems tests come after the completion of unit tests. In "Systems Test in Database Environment" on page 20, the database is restored to use as the starting point for the execution of **PGM-BC** (the *baseline program*). Then the database is unloaded to a Flat File and saved for a future compare. The database is restored again for execution of **PGM-BC'** (the *test-line program*). Now COMPAREX compares the Flat File directly against the database.

Variations on the SEGMENT keyword or the KEY keyword are used to show COMPAREX how to synchronize the two databases. The database produced by program PGM-BC is used as SYSUT1, and the database produced by program PGM-BC' is used as SYSUT2.

Again, the Differences Report is the proof of the effectiveness of the program modification. The user reviews the report to reconcile the actual results to the anticipated results. The user makes necessary program corrections, restores the database, and reruns the program and COMPAREX until all differences have been reconciled.

Managing the Testing Function

The programming manager shares, with the functional manager, the responsibility for effective and accurate performance of computerized systems.

Two areas that programming managers find to be especially troublesome are effective communications about systems requirements and effective testing of requested modifications.

By careful management of the testing function, the programming manager can nearly eliminate trouble from these two sources.

The steps to take are to set down requirements in writing, to develop a test plan for each implementation and modification, to gain the approval of the functional department for the test plan, and to gain the approval of the functional department for the test results.

Set Down Requirements In Writing

The programming manager and the functional manager develop, together, a procedure for written communications. This procedure may include a special form that authorizes implementations and modifications; but, most importantly, the procedure states that computerized systems may not be modified until a written communication, signed by a defined authorizer, reaches the programming manager.

Often, it is the programming department who writes the communication, after discussions with the functional user. The programming manager checks to see that the correct authorizer has signed the communication, and the programming manager schedules the work.

Develop A Test Plan

For each implementation and modification, the programming manager directs that a test plan be developed and set down in writing. The test plan must spell out what tests will be run and who will do the work. In addition, the test plan says how the programming department will know that the test was correct.

For example, the payroll manager sends an authorized written communication to the programming manager to add to the payroll, effective the first of next May, a new deduction for union dues of \$20 per pay period for members of Local ABC123.

The deduction of union dues is nothing new; it is only that Local ABC123 has just negotiated the deduction of its dues with corporate management.

Garner Approvals For The Test Plan

The programming manager and the functional manager negotiate the approval of the test plan.

The functional manager may ask for more test data, and the programming manager will estimate the costs, in terms of dollars and schedule impact, of these requests. At some point, the functional manager will agree to the test plan and to the use of his or her resources to meet the requirements of the test plan.

In our payroll and union dues example, the payroll manager probably will want to see a tally of both gross pay and net pay on the new master. In addition, the payroll manager may want to review certain sensitive accounts on the new master. The programming manager adds these items to the test plan, the payroll manager authorizes the time of M. Smith and J. Jones to review results, and both the payroll manager and the programming manager sign the test plan.

APPLICATION PROGRAMMING TEST PLAN

DEPARTMENT: 3426

DATE: March 3, 1987

PROJECT MANAGER: Bill Winston

PACKAGE #: UNN0032

TEST ITEMS:	WHAT TO TEST:	REVIEWERS/APPROVALS:
1) Accounts to be tested for deduction 123-45-6789 234-56-7890 345-67-8901	1) Positions 127-134 = 'ABC123 ' and positions 135-139 = X'000002000C'	1) Mike will review each COMPAREX report.
2) Dollar figure on file for Local ABC123 is \$4200.00	2) Select records where bytes 127-134 contain 'ABC123 '; tally amount in bytes 135-139; no detail.	2) Attach report to File Folder if amount checks out. Otherwise see Mike.
3) Total union dollar figure	3) Tally bytes 135-139 on entire file	3) Subtract \$4200 from total; Jim will compare to payroll register.
4) Compare previous master to new master MAXDIFF=250, CONTINUE MASK=(7,4) - Date	4) Each differing record should be Local ABC123 member.	4) Attach COMPAREX report to File Folder if no unexpected differences; else see Project Manager.

TEST PLAN APPROVAL:

Project Leader _____	Date _____
Programming Mgr. _____	Date _____
Functional Mgr. _____	Date _____

Generating Test Data

The test plan states the conditions to be tested, and these conditions must be present on the data on the test input files. COMPAREX acts as a test data generator by enabling the user to select certain records onto file SYSUT3.

In our payroll and union dues example, the programmer may decide to do his or her initial tests on a file extracted from live production data containing only the three accounts to be tested plus the payroll manager's sensitive accounts. The programmer could select these off the master file with a COPYDIFF run and desensitize certain fields:

```
MAXDIFF=100,CONTINUE,COPYDIFF
SYSUT1=DUMMY
FILTORIN=(9,EQ,X'123456789C')
FILTORIN=(9,EQ,X'234567890C')
FILTORIN=(9,EQ,X'345678901C')
FILTORIN=(9,EQ,X'890123456.')
FILTORIN=(9,EQ,X'901234567.')
DESEN2=(30,C'EMPLOYEE NAME WAS HERE ') /* Desensitizer */
```

When generating test data, it is necessary to select items that should exercise the new code and items that should not. In our payroll example, the test data should include members of Local ABC123, members of other unions where union dues are deducted, members of other unions where union dues are not deducted, and employees who are not members of any union.

Functional Department Approval

The last step in the management of the testing function is to gain the approval of the functional department for the results of the tests.

The test plan, as signed by the functional manager, has specified the tests to be run and the anticipated results of these tests. In addition, functional department personnel have been assigned to review test data.

Anticipated Results Are Met

If the anticipated results are met, the programmer notes that the results checked out as he or she places the proof of the results in the project folder.

In our example, the second item on the test plan specifies that the dollar figure on the file for Local ABC123 must equal \$4200. If that exact figure is tallied, the programmer places the report in the project folder as proof that the figure was met and he or she makes a note about this test in the column for test approval.

Anticipated Results Are Not Met

If the anticipated results are not met, the programmer must seek help, from the project leader or from the functional user, to reconcile the differences.

In our example, the second item on the test plan specifies that the dollar figure on the file for Local ABC123 must equal \$4200. If that exact figure is not tallied, the programmer is directed to M. Smith for resolution. M. Smith may find that some employees were coded for Local ABC123 in error or that some Local ABC123 members were not coded or that \$4200 is the wrong number. M. Smith, the functional department's employee, has been assigned the task of reconciling these numbers by the test plan; and the programmer and M. Smith will work together to resolve the problem.

Specifications Are Wrong

While the test approvals are being secured, either by computerized checks or by manual lookups, the correctness of the original specifications is tested.

In our example, the functional user may bring to light the knowledge that some Local ABC123 members are coded as ABC12-3, due to an earlier confusion about the designation of the organization.

The programming department and the functional department work together to get the project accomplished. By the time M. Smith and J. Jones sign the test approval, any specifications errors have been corrected, and the implementation of the change will be error free.

Chapter 3 - General

Legend

This legend describes the symbols and abbreviations used in the descriptions of the COMPAREX keywords in the chapters that follow. These symbols and abbreviations are used in this same way on the COMPAREX User Reference Card and in the COMPAREX Quick Start Manual.

- [] Brackets enclose an optional entry.
- () Parentheses must be coded as shown in the examples.
- { } Braces indicate a required entry when more than one selection is available.
- CAPS** Uppercase letters indicate a keyword, name, or field to be coded as shown.
- lowercase** Lowercase letters indicate that variable information is to be supplied.
- underscore Underscores indicate the default value.
- ddd** Displacement. Values range from 1 to 32768 for the default mode, or values range from 0 to 32767 if **MODE=SYSTEMS** is specified.
- lll** Length, in bytes. Values range from 1 to 32767 (*for KEY and KEY1, values range from 1 to 256*).
- t** Type. Values are **X** for hexadecimal and **C** for character.
- vvv** Literal value between apostrophes. e.g. t'vv' could be X'5B'.

Sample Execution

Sample execution JCL is shown in "Sample Batch JCL" on page 25. "Sample Foreground CLIST" on page 25 illustrates foreground execution through a TSO CLIST. TSO users will probably prefer to use the ISPF interface, see "ISPF Interface" on page 54.

```
//OS$JOB      JOB ...
//JOBLIB      DD DISP=SHR,DSN=...
//*
//COMPARE PROC
//COMPAREX EXEC PGM=COMPAREX,
//              REGION=200K
//SYSPRINT DD SYSOUT=*
//              PEND
//*
//EXAMPL01 EXEC COMPARE
//SYSUT1      DD DISP=SHR,DSN=...
//SYSUT2      DD DISP=SHR,DSN=...
//* Optional //SYSUT3 goes here
//SYSIN       DD *
*** COMPAREX keywords go here
/*
```

Figure 11. Sample Batch JCL

```
PROC 2 DSNSYSUT1 DSNSYSUT2
FREE  FI(SYSUT1,SYSUT2,SYSPRINT,SYSIN)
ALLOC FI(SYSPRINT) DA(*)
ALLOC FI(SYSUT1) DA(&DSNSYSUT1) SHR
ALLOC FI(SYSUT2) DA(&DSNSYSUT2) SHR
WRITE *****
WRITE * PREPARE TO CREATE SYSIN INPUT TO *
WRITE * PROGRAM COMPAREX. TERMINATE *
WRITE * SYSIN WITH A /* (SLASH-ASTERISK). *
WRITE * KEY IN HELP FOR ASSISTANCE. *
WRITE *****
ALLOC FI(SYSIN) DA(*)
CALL 'somnode.COMPAREX.LOADLIB(COMPAREX)'
```

Figure 12. Sample Foreground CLIST

Keyword General Comments

COMPAREX keywords are free-form, and they can appear in any order. Here, however, are some rules:

1. All displacements are relative to one (unless the user specifies `MODE=SYSTEMS` to set displacements relative to zero). These displacements occur in `FIELD`, `FILTER`, `IDENTITY`, `MASK`, `KEY`, `DESEN` and `SEGMENT` statements. This means that the first byte of a record is byte number one. For example, if a key in the file is the account number and it occurs in the first four bytes of the record, the `KEY` statement would look like:

`KEY=(1 , 4)`

2. No spaces are allowed within a keyword and its associated data, but a space may be used to separate keywords on the `SYSIN` record.
3. Displacement and length values greater than 32768 are set to 32768.
4. If a file is variable length (`RECFM=V` or `VB`), the `LLBB` (or `RDW`) is not able to be accessed unless `MODE=SYSTEMS` has been specified. See the description of the `MODE` keyword in "Chapter 5 - Input Processing Keywords" on page 69.
5. Commas and decimal points cannot be used in numbers in keywords. Values, displacements, and lengths are given in numeric characters only.
6. Numeric expressions are specified with one to eight numeric characters only. Leading zeroes are acceptable but are not redisplayed in acknowledgement messages.
7. If the first position of a `SYSIN` record is an asterisk, COMPAREX considers the entire record to be a comment and it does not search for keywords on that record.
8. As many keywords as possible may be coded on one `SYSIN` record.
9. Or, each `SYSIN` record may have only one keyword.
10. But, no keyword may extend from one `SYSIN` record to the next.
11. The user may enter as many of each of the following keywords as desired, but COMPAREX will use only the last one specified:
 - a) `BUFF`
 - b) `COPYDIFF`
 - c) `CPXIFACE`
 - d) `DASH`
 - e) `DELETE`
 - f) `DIRECTORY`
 - g) `FORMAT`
 - h) `HALT`
 - i) `IGNORSIN`
 - j) `INSERT`
 - k) `KILLRC`
 - l) `LINE`
 - m) `MAXDIFF`
 - n) `MBRHDR`
 - o) `MLC`
 - p) `MODE`
 - q) `PAGE`

- r) PLUS
- s) REPLACE
- t) SKIPUT1
- u) SKIPUT2
- v) STOPAFT
- w) SYSUT1
- x) SYSUT2
- y) SYSUT3
- z) TEXT
- aa) WILDCARD

12. Some keyword pairs are mutually exclusive. If both are specified, only the last one is used. These pairs are:

- a) DATA versus TEXT versus DIRECTORY
- b) DECIMAL versus HEX
- c) EBCDIC versus ASCII
- d) MODE=APPLICATIONS versus MODE=SYSTEMS
- e) PRINT=MATCH versus PRINT=NOMATCH
- f) PRINT=MISMATCH versus PRINT=NOMISMATCH

13. Some keywords are cumulative. COMPAREX will use as many as the user enters, up to some limit. These keywords are:

- a) Up to 400 IDENTITYs, FIELDs, MASKs, and DESENs are used together by COMPAREX. Each FIELD1 and FIELD2 pair counts as one FIELD and each MASK1 and MASK2 pair counts as one MASK.

In addition, COMPAREX allows for a table of 2020 bytes to hold all IDENTITYs and DESENs. Users entering more than 60 IDENTITY-DESEN combination keywords should read the information on the calculation of the IDENTITY table space in "Chapter 5 - Input Processing Keywords" on page 69.

- b) COMPAREX allows for a table of 2020 bytes to hold all FILTERs. Users entering more than 60 FILTER keywords should read the information on the calculation of the FILTER table space in "Chapter 5 - Input Processing Keywords" on page 69.
- c) Up to 40 KEY statements may be used. Each KEY1 and KEY2 pair counts as one KEY statement.
- d) COMPAREX allows for a table of 1024 bytes to hold all SEGMENT keywords. Users entering more than 30 SEGMENT keywords should read the information on the calculation of the SEGMENT table space in "Chapter 6 - Data File Synchronization Keywords" on page 99.
- e) Up to 40 SQUEEZE statements may be entered.

14. If the user misspells a keyword or incorrectly supplies a variable, COMPAREX underscores the entry with the DASH character and displays the literal "ERROR?" to the right of the underscores on the Differences Report. See "Chapter 10 - Messages" on page 148 message CPX00I, for more information about correction of keywords.

15. COMPAREX will determine the dataset organizations of the files to be processed. The SYSUT1, SYSUT2, and SYSUT3 keywords entered can override the dataset organization determined by accessing control blocks through the operating system, but it is not recommended.

For access to PANVALET, LIBRARIAN, and OTHER proprietary file structures, exit information must be provided to the COMPAREX interface (CPXIFACE) via the SYSUT1 and SYSUT2 keywords.

Chapter 4 - Interfaces

COMPAREX interfaces to many (via CPXIFACE) different data collection structures such as PANVALET, LIBRARIAN, DL/1, IDMS etc. The success of these direct interfaces is very often dependent on how CPXIFACE was generated. If you experience difficulties, first contact your systems programmer who installed COMPAREX, to see how he/she generated it. As additional diagnostic information, you may inspect message CPX20I to see the INFO feedback of its generation. There is a friendly interface from the user to COMPAREX called the ISPF interface, see "ISPF Interface" on page 54.

PANVALET, LIBRARIAN, and GEM

COMPAREX can interface directly to Pansophic's PANVALET, Applied Data Research's LIBRARIAN, or FUJITSU/FACOM's GEM.

PANVALET

Refer to "Direct PANVALET Interface" below for an example of comparing two members on the same PANVALET library and generating an audit trail for subsequent browsing and/or feeding back into PAN#1.

```
//COMPARE EXEC PGM=COMPAREX,REGION=512K
//SYSPRINT DD SYSOUT=*
//PANDD1 DD DISP=SHR,DSN=somnode.PANLIB
//SYSUT3 DD DISP=(,PASS),DSN=&&AUDIT,
// UNIT=SYSDA,SPACE=(TRK,(1,1)),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
//SYSIN DD *
    SYSUT1=(PAN,MEMBER=cobolname1) /* Fill in cobolname1 */
    SYSUT2=(PAN,MEMBER=cobolname2) /* Fill in cobolname2 */
    TEXT=COBOL
    COPYDIFF=(PAN,STAMP=YES) /* Generate Audit Trail */
//*
//TRAIL EXEC PAN$PROC <== Invokes PGM=PAN#1
//PANDD1 DD DISP=SHR,DSN=somnode.PANLIB
//SYSIN DD DISP=(OLD,DELETE),DSN=&&AUDIT
//* EOJ
```

Figure 13. Direct PANVALET Interface

Refer to "Direct PANVALET Interface" on page 29 for an example of comparing a member of one PANVALET library against a member of another PANVALET library.

```

//COMPARE EXEC PGM=COMPAREX,REGION=512K
//SYSPRINT DD SYSOUT=*
//PANDD1 DD DISP=SHR,DSN=somnode.PANLIB
//PANLIB2 DD DISP=SHR,DSN=somnode.PANLIB2 <=== Note
//SYSIN DD *
    SYSUT1=(PAN, MEMBER=cobolname1) /* Fill in cobolname1 */
    SYSUT2=(PAN, MEMBER=cobolname1, DDNAME=PANLIB2) /* <=== Note */
    TEXT=COBOL
// * EOJ

```

Figure 14. Direct PANVALET Interface

Potential Error Messages

- PAN - MEMBER NOT FOUND
- PAN - RC=xxxx: last successfully read record (first 60 bytes)

LIBRARIAN

Refer to "Direct LIBRARIAN Interface" on page 29 for an example of comparing a member of a LIBRARIAN Master against a sequential dataset.

Note: Your LIBRARIAN release level must be 3.2 or higher. Release 3.1 will not work at all.

```

//COMPARE EXEC PGM=COMPAREX,REGION=512K
//SYSPRINT DD SYSOUT=*
//MASTER DD DISP=SHR,DSN=somnode.LIBRARIAN.MASTER
//SYSUT2 DD DISP=SHR,DSN=somnode.PROGRAM1.ASM
//SYSIN DD *
    SYSUT1=(LIB, MEMBER=PROGRAM1)
    TEXT=BAL
// * EOJ

```

Figure 15. Direct LIBRARIAN Interface

Refer to "Direct LIBRARIAN Interface" on page 30 for an example of comparing the directories of two LIBRARIAN Masters.

ADABAS

ADABAS from Software AG is a DBMS that is not hierarchical. Refer to "ADABAS Interface Record Layout" below for a graphical layout of what the database records look like as returned to COMPAREX by CPXIFACE. If you are accessing *Natural* programs, there is no I.S.N. returned.

Bytes	-- C o n t e n t s --
1-4	I.S.N. of returned record
5-n	Returned Data, variable length

Figure 18. ADABAS Interface Record Layout

See "Direct ADABAS Interface" below for an example of the keyword structure necessary to read ADABAS data files.

```
//ADABAS EXEC PGM=COMPAREX,REGION=512K
//STEPLIB DD DISP=SHR,DSN=somnode.COMPAREX.LOADLIB
//SYSPRINT DD SYSOUT=*
//* <=== ADABAS specific DD card images go here.
//SYSIN DD *
  CPXIFACE=CPXADABS /* Special interface module */
*****
* Point SYSUT1 to the Database 3, File 4, Logical key field BC *
* and retrieve fields AA, AB, AU, BF, and DX1-5. *
*****
  SYSUT1=(OTH,PARM='D03F4,K=BC,F=BC,AA,AB,AU,BF,DX1-5')
*****
* Point SYSUT2 to the Database 2, File 9, Logical key field BJ *
* and retrieve fields CA, CB, CU, DF, and FX1-5. *
*****
  SYSUT2=(OTH,PARM='D2F9,K=BJ,F=BJ,CA,CB,CU,DF,FX1-5')
  KEY=(1,8,,A),FORMAT=25,MAXDIFF=50,CONTINUE
//* EOJ
```

Figure 19. Direct ADABAS Interface

The key to the specifications here is within the PARM= subparameter of SYSUT1 and SYSUT2. The format of the "PARM" information is:

```
PARM=Ffff
or
PARM='DdddFfff,K=kk,F=bbbbbbbbbbbbbbbbbbbbbb'
or
```



```
PARM= 'DddFff,LS=11111111'
```

or

```
PARM= 'Ff,LO=11111111'
```

where apostrophes are mandatory if any commas or blanks are present (normal case) within the parameter data.

The MEMBER (*or M*) option is used for specifying particular members of Natural libraries, either source or object.

In reading DATA, a read physical (L2) call is the default unless you want a read logical (L3) call, e.g.

```
PARM= 'D3F12,K=AB'
```

which will read database 3 and file 12 logically in ascending sequence of field 'AB'. You may also specify the particular fields to be retrieved. If you don't specify fields, all fields will be retrieved of course. The field specifications are exactly that for the Format Buffer which you should be familiar with, e.g.

```
PARM= 'D001F013,K=AR,F=AR,AA,AB,AC'
```

In reading Natural programs, use the LS= (*Library of Source*) or LO= (*Library of Object*) options, e.g.

```
SYSUT1=(OTH,M=MEMBER1,PARM='D1F8,LS=MYSOURCE')
```

or

```
SYSUT2=(OTH,PARM='D2F8,LO=MYOBJECT')
```

In the COMPAREX tradition, individual members or entire libraries (*subject to Filtering of course*) can be processed.

Potential Error Messages

- ADABAS - OPEN ERROR - xxx
 - In the special case of xxx = 152 or 053; the user must modify the "LU" keyword parameter of "ADARUN" to increase (recommend LU = 60000) the size.
- ADABAS - PARAMETER DATA ERROR
- ADABAS - DATABASE/FILE ERROR
- ADABAS - FIELD DEFINE ERROR - xxx
- ADABAS - LOGICAL READ (L3) ERROR - xxx
- ADABAS - MEMBER NOT FOUND
- ADABAS - READ ERROR - xxx

Condor CAMLIB

This read-only interface processes the library structure known as "CONDOR CAMLIB." You may compare the directories, individual members or the entire library to a PANVALET, LIBRARIAN, or another CAMLIB library.

Invoking COMPAREX to Process a CAMLIB

See "Direct CAMLIB Interface" on page 35 for an example of how to compare the directories of one CAMLIB against another.

The default filename for SYSUT1 is GAUT1 (*GAUT2 for SYSUT2*). You may specify another filename in this format:

```
SYSUT1=(OTH,DDNAME=ABC1)
```

but only the first four characters will be used and it will be internally prefixed with GA by the CAMLIB module USROPN.

If you are after a particular member and it has multiple versions and/or a password, you must specify the particular version and/or its associated password in the "PARM" area:

```
SYSUT1=(OTH, MEMBER=MEMBER, PARM='002WXYZ')
```

where the layout of PARM is fixed in this fixed format:

```
PARM='nnnpass'
```

where 'nnn' is three numeric digits and 'pass' is the password.

Speaking of passwords, if you compare an entire CAMLIB library to some other library (another CAMLIB perhaps) and you match on a member name that is indeed password protected, the actual content of the member will not be read and compared. Instead, a single record will be returned with the following contents:

```
--PASSWORD PROTECTED/MEMBER SKIPPED-- 1
```

informing the user that this password protected member is intentionally being bypassed. On your differences report (assuming TEXT=COBOL) this single line will appear where the SYSUT1 member would normally appear. Similarly,

```
--PASSWORD PROTECTED/MEMBER SKIPPED-- 2
```

will appear where the SYSUT2 member would normally appear.

```
//yourjob JOB ...  
//COMPAREX EXEC PGM=COMPAREX,REGION=256K  
//SYSPRINT DD SYSOUT=*  
//SYSUT1 DD DISP=SHR,DSN=somnode.FIRST.CAMLIB  
//SYSUT2 DD DISP=SHR,DSN=somnode.SECOND.CAMLIB  
//SYSIN DD *  
CPXIFACE=CPXCAMLB /* Special interface module */  
SYSUT1=OTH,SYSUT2=OTH  
DIR=PDF
```

Figure 20. Direct CAMLIB Interface

Potential Error Messages

- CAMLIB - USROPN ERROR - xx
- CAMLIB - USRDIR ERROR - xx
- CAMLIB - MEMBER NOT FOUND
- CAMLIB - PASSWORD MISMATCH
- CAMLIB - USRGET ERROR - xx

DL/1

DL/1 databases are hierarchical in structure. IBM, the developer of DL/1, has coined the terms HDAM, HIDAM, HISAM, and SHISAM to talk about the organizational methods, but all these methods are essentially usages of QSAM, ISAM, and VSAM. These QSAM, ISAM, and VSAM file organizations can be handled directly by COMPAREX.

There are ways to unload a DL/1 database to a flat file other than COMPAREX, but they are not recommended in preparation for a comparison. COMPAREX will not logically understand an "Image Copy" of a DL/1 database. Refer to "DL/1 Interface Segment Layout" below for a graphical layout of what the individual records look like as returned to COMPAREX by CPXIFACE.

Bytes	-- C o n t e n t s --
1-8	Segment Name padded with blanks
9-64	Concatenated Key padded with nulls
65-n	Returned Segment Data, variable length

Figure 21. DL/1 Interface Segment Layout

It sometimes is necessary to unload both of the databases to flat files and sort them before having COMPAREX compare them. One of the cases where this is advised is when the databases are large HDAM files and either the randomizing algorithm has changed or there are many inserts and/or deletes. If sorting is required, sort on the concatenated key in ascending order.

You may use any PSB (except one that has Proc Option "L" for load) that exists in your applicable PSBLIB so long as it contains a PCB window for the appropriate DBD - database. If you want to compare two databases in the same execution, the PSB must contain at least two PCB's with unique DBDname's. For FASTPATH, set KW=IFP and for full function, set KW=DLI. Always set PROG=COMPAREX.

To avoid searching PSBLIB's for usable PSB's etc, it is strongly recommended that the shop's DataBase Administration (DBA) group create a large PSB called COMPAREX that contains two PCB's for every database in the shop with all proc options GO - get only. The reason for two PCB's per database is that a user may want to compare two versions of a database - one real and the other an alias. This requires that every DBD be in the DBDLIB twice, once for itself and once for the alternate name.

Refer to "Direct DL/1 Interface" on page 37 for an example of invoking COMPAREX under DL/1 to read the databases directly in two passes.

Note: It is not possible to use the ISPF Interface to read DL/1 databases directly.

```

//DLISOS  PROC KW=DLI,SIZE=2048,PROG=,PSB=, etc.
//DFSRR00 EXEC PGM=DFSRR00,REGION=&SIZE.K,
//          PARM='&KW,&PROG,&PSB, etc'
//STEPLIB  DD DISP=SHR,DSN=somnode.IMSVS.RESLIB
//          DD DISP=SHR,DSN=somnode.COMPAREX.LOADLIB
//* ... (Other DD card images pertinent to DLI, Databases)
//          PEND
//*
//UNLOAD  EXEC DLISOS,PROG=COMPAREX,SIZE=1024,
//          PSB=psbname,KW=DLI <=== or KW=IFP
//SYSPRINT DD SYSOUT=*
//SYSUT3   DD DISP=(,CATLG,DELETE),DSN=somnode.DLIBASE,
//          UNIT=SYSDA,SPACE=(TRK,(9,9),RLSE),
//          DCB=(RECFM=VB,LRECL=4000,BLKSIZE=4096)
//SYSIN    DD *
//          CPXIFACE=CPXDLI          /* Special generation */
//          SYSUT1=DUMMY
//          SYSUT2=(OTH,MEMBER=dbdname) /* <=== Fill in 'dbdname' */
//          COPYDIFF,MAXDIFF=5,CONTINUE /* No need to print it all */
//*
//* At this point, we have unloaded the database to a flat file
//* called "somnode.DLIBASE" Now we can optionally restore the
//* database and update it iteratively with whatever programs
//* are being tested. Then we COMPAREX the flat file against
//* the freshly updated database to see what changes were made.
//*
//COMPARE EXEC DLISOS,PROG=COMPAREX,
//          PSB=psbname,KW=DLI <=== or KW=IFP
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DISP=SHR,DSN=somnode.DLIBASE
//SYSIN    DD *
//          MAXDIFF=50,CONTINUE      /* Generally advised */
//          CPXIFACE=CPXDLI          /* Special generation */
//          SYSUT2=(OTH,MEMBER=dbdname) /* <=== Fill in 'dbdname' */
//          KEY=(1,64,,R),BUFF=1024 /* Random KEY, large BUFFER */
//* EOJ

```

Figure 22. Direct DL/1 Interface

Refer to "Direct DL/1 Interface" on page 38 for an example of invoking COMPAREX under DL/1 to compare two databases directly in a single pass.

```

//DLISOS  PROC KW=DLI,SIZE=2048,PROG=,PSB=, etc.
//DFSRR00 EXEC PGM=DFSRR00,REGION=&SIZE.K,
//          PARM='&KW,&PROG,&PSB, etc'
//STEPLIB  DD DISP=SHR,DSN=somnode.IMSVS.RESLIB
//          DD DISP=SHR,DSN=somnode.COMPAREX.LOADLIB
//* ... (Other DD card images pertinent to DLI, Databases)
//          PEND
//*
//COMPARE EXEC DLISOS,PROG=COMPAREX,
//          PSB=COMPAREX          <=== Note PSB Name
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
          CPXIFACE=CPXDLI          /* Special generation */
          SYSUT1=(OTH,MEMBER=dbdname1) /* <=== Fill in 'dbdname1' */
          SYSUT2=(OTH,MEMBER=dbdname2) /* <=== Fill in 'dbdname2' */
          MAXDIFF=50,CONTINUE      /* Generally advised */
          KEY=(1,64,,R),BUFF=1024  /* Random KEY, large BUFFER */
//* EOJ

```

Figure 23. Direct DL/1 Interface

Note: The ISPF interface will not work with this interface. Too many special DD cards are needed.

Potential Error Messages

- DL/1 - CANNOT FIND DBDNAME IN PSB
- DL/1 - READ ERROR,FUNC=GN,STATUS=xx

DMS - DASD Management System

DMS is a system of migrating, *or archiving*, datasets from DASD to less expensive storage media. It can also be used to back up certain large datasets in case they are crucial. This is the where this interface comes into play. It can be used to determine how much (*percentage*) a large VSAM dataset has changed since it was backed up.

See "Direct DMS Interface" below for an example of how to compare an archived VSAM file against the current VSAM file.

```
//yourjob JOB ...
//COMPAREX EXEC PGM=COMPAREX,REGION=256K
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DSN=somnode.DMSC.D16MAY86.T140716
//SYSUT2 DD DISP=SHR,DSN=somnode.VSAM.FILE
//SYSIN DD *
*****
* Point SYSUT1 to the archived VSAM file - may be on tape *
*****
SYSUT1=(OTH,M=ANYTHING,PARM='somnode.VSAM.FILE')
KEY=(4,9),FORMAT=06,MODE=SYSTEMS
/*
```

Figure 24. Direct DMS Interface

The PARM must specify the full dataset name.

Potential Error Messages

- DMS - OPEN ERROR
- DMS - NOT A DMS FORMATTED BACKUP
- DMS - UNABLE TO FIND REQUESTED DATASET NAME

Disclaimer

The following restrictions apply:

- The archived dataset must be the first dataset on the tape.
- Only archived VSAM or QSAM files will work.
- The archived dataset must NOT be compacted.

IAM

IAM stands for Innovation Access Method from Innovation Data Processing. It is intended to be an ISAM replacement - not a DBMS.

Refer to "Direct IAM Interface" below for an example of invoking COMPAREX with IAM to compare an ISAM file against an IAM file.

```
//ISAM$IAM EXEC PGM=COMPAREX,REGION=256K
//STEPLIB DD DISP=SHR,DSN=somnode.COMPAREX.LOADLIB
// DD DISP=SHR,DSN=somnode.IAM.LOADLIB
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DSN=somnode.ISAM.FILE
//SYSUT2 DD DISP=SHR,DSN=somnode.IAM.FILE
//SYSIN DD *
*****
* The Ascending KEY will be dynamically extracted from the ISAM
* file and used for synchronizing - see message CPX23I.
*****
CPXIFACE=CPXIAM /* Special generation */
SYSUT2=(OTH, MEMBER=anything) /* Must specify a MEMBER */
MAXDIFF=50,CONTINUE
//* EOJ
```

Figure 25. Direct IAM Interface

The MEMBER (or M) option is necessary because COMPAREX assumes that any use of PAN, LIB or OTH is for directory-embedded datasets. In this case "OTH" is not a library but is a sequential file. We still must placate COMPAREX, otherwise we terminate with message:

CPX31A - FILE ORGANIZATIONS NOT COMPATIBLE - . . .

Potential Error Messages

- IAM - OPEN ERROR - I/xxx
- IAM - OPEN ERROR - O/xxx

IDMS

IDMS from Cullinet is a DBMS that is not hierarchical. Both the conventional release (5.7) and the relational release (IDMS/R, release 10.0) are supported. Refer to "IDMS Interface Record Layout" below for a graphical layout of what the individual records look like as returned to COMPAREX by CPXIFACE.

Bytes	-- C o n t e n t s --
1-4	DBKEY of Record
5-n	Returned Data, variable length

Figure 26. IDMS Interface Record Layout

See "Direct IDMS Interface" below for an example of the keyword structure to sweep through two areas of Subschema 'DEMOSS01'. This Subschema is delivered with all releases of IDMS as a post-installation test.

```
//IDMS      EXEC PGM=COMPAREX,REGION=512K
//STEPLIB   DD DISP=SHR,DSN=somnode.COMPAREX.LOADLIB
//          DD DISP=SHR,DSN=somnode.SUBSCHEM.LOADLIB
//SYSPRINT  DD SYSOUT=*
//* <=== IDMS specific DD card images go here.
//SYSIN     DD *
    CPXIFACE=CPX$IDMS /* Special interface module */
*****
* Point SYSUT1 to the Subschema (in two parts) and the      *
* 'Customer' area for the sweep.                             *
*****
    SYSUT1=(OTH,M=DEMOSS01) /* SUBSCHEMA
    SYSUT1=(OTH,PARM='P=COMPAREX,R=CUSTOMER,A=CUSTOMER-REGION')
*****
* Also point SYSUT2 to the Subschema and the 'Order' area for *
* the sweep.                                                  *
*****
    SYSUT2=(OTH,M=DEMOSS01,PARM='R=ORDOR,A=ORDER-REGION')
*****
* These two areas have nothing in common that can be compared *
* for any logical reason. Just see if they can be processed.  *
*****
    MAXDIFF=10,CONTINUE
/* EOJ
```

Figure 27. Direct IDMS Interface

Note: The ISPF interface will not work with this interface. Too many special DD cards are needed.

In addition to the *Area Sweep within record type*, a new facility is also available. There are now two major ways to process the database. The first is a slight variation on the area sweep. The second is to specify an exit module (*written in your own fluent language*) to navigate through the database.

The reason for this is that there are many ways to read through a database, and leaving the exit in the user's hands allows him/her to have more control over what is read and how.

Parameter Specifications

The key to the specifications here is within the PARM= subparameter of SYSUT1 and SYSUT2. The format of the "PARM" information is:

```
PARM='P=ppp,R=rrr,A=aaa'
or
PARM='R=rrr,A=aaa'
or
PARM=EXIT
```

where apostrophes are mandatory if any commas or blanks are present within the parameter data.

If **PARM=EXIT** has been specified, an exit module is called from CPXIFACE to do the actual **OPEN**, **READ**, and **CLOSE**.

To sweep an area (physically), the keyword notation of the PARM is in the format:

```
PARM='P=program,R=record,A=area'
```

where **P** stands for program and is optional. If omitted, the program name used in the IDMS control block will be **COMPAREX**. **R** stands for record name and is required. **A** stands for area name and is also required. Failure to specify a legitimate record name and/or area name will result in a **BIND** error as a minimum.

The returned record now has the **DBKEY** (a fullword - 4 bytes) prefixed so it can be seen (and compared).

If the user has specified that an exit is to be called to navigate through the database, then the **MEMBER** keyword points to the exit module name:

```
SYSUT1=(OTH, MEMBER=JOSEFINE, PARM=EXIT)
```

A sample COBOL exit module called **JOSEFINE**² is given in "IDMS COBOL Exit Module JOSEFINE - Excerpts" on page 45. This program has one entry point but performs the basic functions of Open, iterative Read, and Close based on the request (IFACE-REQUEST) passed in its linkage section from the CPXIFACE module (CPX\$IDMS). This module exhausts (until DB-END-OF-SET) Customers and Orders within those Customers. The data passed back to **COMPAREX** is grouped in exactly the same way as the

```
COPY IDMS SUBSCHEMA-RECORDS.
```

generates them in the linkage section. The length of this data is calculated by CPX\$IDMS (or whatever module name is chosen for CPXIFACE) by walking backward from the maximum size (32K) until the first non-null byte is found. The first three words of the returned area are the DB-KEYs of the component records (we use two of these three words in this example).

The COBOL program given is meant to be a sample. You are encouraged to modify it to navigate differently and/or to work with other Subschemas besides 'DEMOSS01'.

Comparison is usually accomplished by unloading a database first to SYSUT3, then updating the database by whatever method is to be tested, and then comparing the unloaded flat file directly to the database. The iterative process of restoring the database, retesting the TESTLINE code, and comparing for unexpected changes is accelerated. See "Direct IDMS Interface Via Exit Module - Unload" on page 43 for an example of unloading the database. Note that we specified 'MAX-

² Named after a lady named Josephine who assisted in the development

DIFF=5,CONTINUE' in the unload step. We can print the first five records (or more if you wish) and inspect them for FIELDS and synchronizing KEYS. Using the displacement issued on the left of the Differences Report, we can calculate an accurate FIELD/KEY strategy.

The synchronizing strategy is dependent on:

- how many record types are gathered together in a single read
- how you want to treat inserted/deleted high-order records
- how you want to treat inserted/deleted subordinate records

See "Direct IDMS Interface Via Exit Module - Compare" on page 44 for an example of comparing the flat file against the updated database.

Warning

In determining what to call an exit program that works with a particular Subschema,

NEVER NAME THE PROGRAM THE SAME AS THE SUBSCHEMA

The reason for this is that when the exit program is loaded (by CPXIFACE or CPX\$IDMS) into storage for subsequent calling, if it is named the same as any existing Schema or Subschema and the STEPLIB concatenation has the IDMS control blocks higher than the exit module and you will be loading (and executing) a Subschema which gives unpredictable results. Conversely, bringing in the exit module means that the Subschema cannot be loaded by IDMS.

```
//UNLOAD EXEC PGM=COMPAREX,REGION=512K
//STEPLIB DD DISP=SHR,DSN=somnode.IDMS.LOADLIB
// DD DISP=SHR,DSN=somnode.COMPAREX.LOADLIB
//SYSPRINT DD SYSOUT=*
//SYSUT3 DD DISP=(,CATLG,DELETE),DSN=somnode.DEMOSS01.UNLOAD,
// UNIT=3350,SPACE=(TRK,(9,9),RLSE),
// DCB=(RECFM=VB,LRECL=8000,BLKSIZE=8008)
//SYSIN DD *
CPXIFACE=CPX$IDMS /* SPECIAL JUST FOR IDMS */
SYSUT1=DUMMY
SYSUT2=(OTH,M=JOSEFINE,PARM=EXIT)
MAXDIFF=5,CONTINUE /* NO NEED TO PRINT IT ALL
COPYDIFF
```

Figure 28. Direct IDMS Interface Via Exit Module - Unload

```

//COMPARE EXEC PGM=COMPAREX,REGION=768K
//STEPLIB DD DISP=SHR,DSN=somnode.IDMS.LOADLIB
// DD DISP=SHR,DSN=somnode.COMPAREX.LOADLIB
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DSN=somnode.DEMOSS01.UNLOAD
//SYSIN DD *
CPXIFACE=CPX$IDMS /* SPECIAL JUST FOR IDMS */
SYSUT2=(OTH,M=JOSEFINE,PARM=EXIT)
MAXDIFF=100,CONTINUE
KEY=(3397,10,,R) /* CUSTOMER NUMBER
* KEY=(1,12,,R) /* RANDOM KEY ON FIRST 3 DB-KEYS
BUFF=512 /* LARGER BUFFER NECESSARY
FORMAT=06

```

Figure 29. Direct IDMS Interface Via Exit Module - Compare

Potential Error Messages

- IDMS - PARAMETER DATA ERROR
- IDMS - BIND SUB-SCHEMA - xxx
- IDMS - BIND RECORD - xxx
- IDMS - READY ALL - xxx
- IDMS - OBTAIN NEXT RECORD - xxx
- IDMS - ACCEPT DB-KEY - xxx
- IDMS - exitname/{error message from exit module}

```

IDENTIFICATION DIVISION.
*DMLIST.
PROGRAM-ID.      JOSEFINE.
AUTHOR.          SERENA CONSULTING.
DATE WRITTEN.    AUGUST, 1986.
DATE COMPILED.
REMARKS.
*****
*   THIS PROGRAM IS INTENDED TO BE THE MODEL FOR OTHER
*   EXIT MODULES WHEN THE INTENT IS TO HAVE COMPAREX CALL
*   THIS MODULE (THROUGH CPXIFACE) TO READ PROPRIETARY
*   DATA BASE MANAGEMENT SYSTEMS (DBMS).  THIS PARTICULAR
*   MODEL IS FOR:
*
*           IDMS
*
*   NOTE THE LINKAGE-SECTION AND CALL STRUCTURE.  THERE
*   IS A SINGLE ENTRY POINT BUT THE PARAMETER LIST CONTAINS
*   THREE AREAS:
*
*       1) IFACE-REQUEST: 'OPEN', 'READ', OR
*           'CLOS' PLUS FREE-FORM INSTRUCTIONS;
*
*       2) IFACE-RESPONSE: SPACES, 'EOF', OR
*           A LITERAL ERROR MESSAGE;
*
*       3) INTERFACE-RECORD-AREA: LAYOUTS ARE
*           MEANT TO BE COPIED HERE IN WHATEVER
*           FASHION NECESSARY.  WHEN THE RECORD(S)
*           ARE READ AND PASSED BACK TO CPXIFACE,
*           HE CALCULATES THE RECORD LENGTH BY
*           WALKING BACKWARDS FROM THE END OF THE
*           AREA UNTIL THE FIRST OCCURRENCE OF A
*           NON-NULL (NOT X'00') CHARACTER.  THE
*           MAXIMUM SIZE OF THIS AREA IS 32K.
*
*   THE CALL STRUCTURE IS DRIVEN BY THE CONTENTS OF
*   'IFACE-REQUEST':
*
*       OPEN: INVOKES 1000-OPEN-DATABASE;
*       READ: INVOKES 2000-READ-NEXT-RECORD
*             ITERATIVELY UNTIL A NON-BLANK
*             RESPONSE IS RETURNED;
*       CLOS: INVOKES 9000-CLOSE-DATABASE.
*****

```

Figure 30. IDMS COBOL Exit Module JOSEFINE - Excerpts

ISPF Packed File

ISPF, from IBM, is a very sophisticated Library Management System that manages data (usually program source code). It is not a DBMS. Release 2.1.0 features the ability to compress (pack) the PDF edited file. If this packed file is subsequently read by most utilities or compilers, it is unintelligible. COMPAREX can read a packed file and unpack it for comparison through the COMPAREX Interface now. The logic in CPXIFACE allows only a single member at a time and it must be known in advance that the file is indeed packed.

There are plans in the future to have this code implicit in COMPAREX such that it unpacks (or not) the individual member based on the content automatically. At that time, the code will be removed from CPXIFACE.

Refer to "Direct ISPF Packed File Interface" below for an example of invoking COMPAREX to compare an ISPF packed member against an unpacked member.

```
//ISPF PKD EXEC PGM=COMPAREX,REGION=256K
//STEPLIB DD DISP=SHR,DSN=somnode.COMPAREX.LOADLIB
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DSN=somnode.PASCAL.PDS(PASCAL$P)
//SYSUT2 DD DISP=SHR,DSN=somnode.PASCAL.PDS(PASCAL$U)
//SYSIN DD *
CPXIFACE=CPXSPFPK /* Special generation */
SYSUT1=(OTH, MEMBER=PASCAL$P) /* Packed by ISPF 2.1 */
TEXT=PASCAL
//* EOJ
```

Figure 31. Direct ISPF Packed File Interface

Disclaimer

At this writing, release 2.2 of ISPF has changed the packing algorithm. This interface does not yet support this release.

RAMIS II

RAMIS II from Online Software (formerly Martin Marietta - Mathematica Products Group) is a 4th Generation Language (4GL) DBMS that is not hierarchical. Refer to "RAMIS Interface Record Layout" below for a graphical layout of what the individual records look like as returned to COMPAREX by CPXIFACE.

Bytes	-- C o n t e n t s --
1-n	Returned Data, variable length

Figure 32. RAMIS Interface Record Layout

Refer to "Direct RAMIS Interface" on page 47 for an example of invoking COMPAREX with RAMIS to compare two databases directly in a single pass.

```
//RAMIS      EXEC PGM=COMPAREX,REGION=512K
//STEPLIB    DD DISP=SHR,DSN=somnode.COMPAREX.LOADLIB
//           DD DISP=SHR,DSN=somnode.RAMIS.LOADLIB
//SYSPRINT   DD SYSOUT=*
//* <=== RAMIS specific DD card images go here.
//DATABASE   DD DISP=SHR,DSN=somnode.DATABASE
//SYSIN      DD *
    CPXIFACE=CPXRAMIS          /* Special generation */
    SYSUT1=(OTH, MEMBER=filenam1)
    SYSUT2=(OTH, MEMBER=filenam2)
    MAXDIFF=50,CONTINUE
    KEY=(1,nn,,R),BUFF=256 /* Random KEY, large BUFFER */
*           * **=====> Fine tune the KEY specification.
//* EOJ
```

Figure 33. Direct RAMIS Interface

The MEMBER name must specify a valid file (up to 12 characters) name.

Potential Error Messages

- RAMIS II - OPEN ERROR - OPEM/xxxx
- RAMIS II - OPEN ERROR - LOCN/xxxx
- RAMIS II - MEMBER NAME MUST SPECIFY FILENAME
- RAMIS II - READ ERROR - NEXR/xxxx

ROSCOE

ROSCOE, from Applied Data Research, is a Library Management System that manages data (usually program source code). It is not a DBMS. Refer to "Direct ROSCOE Interface" on page 48 for an example of invoking COMPAREX with ROSCOE to compare two members directly.

```
//ROSCOE EXEC PGM=COMPAREX,REGION=256K
//STEPLIB DD DISP=SHR,DSN=somnode.COMPAREX.LOADLIB
// DD DISP=SHR,DSN=somnode.ROSCOE.LOADLIB
//SYSPRINT DD SYSOUT=*
//* <=== ROSCOE specific DD card images go here.
//ROSLIB00 DD DISP=SHR,DSN=somnode.roslib00
//ROSLIB01 DD DISP=SHR,DSN=somnode.roslib01
//ROSLIB02 DD DISP=SHR,DSN=somnode.roslib02
//SYSIN DD *
CPXIFACE=CPXROSCO /* Special generation */
SYSUT1=(OTH,MEMBER=member1,PARM=pfxcd)
SYSUT2=(OTH,MEMBER=member2,PARM=pfxcd)
TEXT=COBOL
//* EOJ
```

Figure 34. Direct ROSCOE Interface

Use the MEMBER options to specify a single member name to be read. To scan the entire library (FILTERs can limit this) leave off the MEMBER option. The PARM must be specified in this format:

```
PARM='xxxxyy'
      |  |
      |  |=====> Two character User Code
      |=====> Three character ROSCOE Prefix
```

Note: The ISPF interface will not work with this interface. Too many special DD cards are needed.

Potential Error Messages

- ROSCOE - INITX ERROR - xxxx
- ROSCOE - ACTIVATE ERROR - xxxx
- ROSCOE - FIND ERROR - xxxx
- ROSCOE - LIB (WALK) ERROR - xxxx
- ROSCOE - MEMBER: member - ACCESS DENIED BY SITE MANAGEMENT!
- ROSCOE - GET ERROR - xxxx

WYLBUR

WYLBUR is a Library Management System that compacts the data (*usually program source code*) into sequential datasets or members of Partitioned DataSets (PDS's). Refer to "Direct WYLBUR Interface" below for an example of invoking COMPAREX with WYLBUR to compare two members directly.

```
//WYLBUR EXEC PGM=COMPAREX,REGION=256K
//STEPLIB DD DISP=SHR,DSN=somnode.COMPAREX.LOADLIB
//SYSPRINT DD SYSOUT=*
//WYLBUR DD DISP=SHR,DSN=WYL.abc.def.LIB(member1)
//SYSUT2 DD DISP=SHR,DSN=WYL.abc.def.member2 <=== Sequential
//SYSIN DD *
CPXIFACE=CPXWYLBR /* Special generation */
SYSUT1=(OTH,M=member1,DDNAME=WYLBUR)
SYSUT2=(OTH,MEMBER=member2)
TEXT=COBOL
//* EOJ
```

Figure 35. Direct WYLBUR Interface

The MEMBER (or M) option is necessary because COMPAREX assumes that any use of PAN, LIB or OTH is for directory-embedded datasets. In this case "OTH" is not a library but is a sequential file. We still must placate COMPAREX, otherwise we terminate with message:

CPX31A - FILE ORGANIZATIONS NOT COMPATIBLE - . . .

Note: The WYLBUR interface does not "walk" library members. It only reads sequential files or individual members of PDS's. If you attempt to have him read a PDS as opposed to a member of a PDS, results are unpredictable.

Potential Error Messages

- WYLBUR - OPEN ERROR
- WYLBUR - NOT IN WYLBUR FORMAT

Roll Your Own

This is where you may code your own proprietary Library/DataBase Management System interface and set your own syntax rules. The source code to CPXIFACE is provided with this slot open. Use the other source code supplied under "OTH", as models for building your own interface.

Refer to "CPXIFACE Sample Code - Roll Your Own Slot" below for an example of the source code changes to CPXIFACE to read a variable length record and *conditionally* explode the record.

```

...
&OTH      SETB  1          (YES)          GENERATE THE 'OTHER' INTERFACE.
...
&ZROLOWN  SETB  1          (YES)          .ROLL YOUR OWN - PROPRIETARY
...
AIF      (NOT &ZROLOWN).ROL3900  ROLL YOUR OWN - READ
...
ROL$3060  LR      R15,R1          DITTO LENGTHS
          STH      R1,PRM$RLEN    PASS LENGTH BACK TO COMPAREX
          MVCL     R0,R14        LARGE MOVE
* (Insert the following lines of code)
          CLI      PRM$WICH,C'1'  SYSUT1
          BE       RETURN        YES, RETURN NOW
          TM       PRM$SWUS,SWPRMSAM  SYSUT1 AND SYSUT2 SAME INTERFACE
          BZ       RETURN        NO, BRANCH
          L        R2,PRM$PS12    SYSUT1'S INTERFACE
          CLC      PRM$RLEN,PRM$RLEN-PARAMTER(R2) SAME LENGTH
          BNE      ROL$3200       NO, EXPLODE THEM BOTH
          L        R0,PRM$ADDR-PARAMTER(R2) SYSUT1'S RECORD ADDRESS
          LH       R1,PRM$RLEN-PARAMTER(R2) SYSUT1'S RECORD LENGTH
          L        R14,PRM$ADDR   SYSUT2'S RECORD ADDRESS
          LH       R15,PRM$RLEN   SYSUT2'S RECORD LENGTH
          CLCL     R0,R14        COMPARE THEM
          BE       RETURN        IDENTICAL, RETURN WITHOUT EXPLODING
ROL$3200  EQU      *            CALL YOUR EXPLOSION ROUTINE FOR BOTH
          L        R14,PRM$ADDR-PARAMTER(R2) SYSUT1'S RECORD ADDRESS
          LA       R15,yourstuf   YOUR SPECIAL EXPLOSION PARAMETER
          ICM      R15,B'1000',LASTLIST INDICATE LAST IN LIST
          STM      R14,R15,PRM$WORK  EXPLoder PARAMETER LIST
*          CALL    exploder,PRM$WORK  CALL YOUR EXPLOSION ROUTINE
          LA       R1,PRM$WORK    +ADDRESS THE ADDRESS LIST
          L        R15,=V(exploder) +<=== Your Explosion Program
          BALR     R14,R15        +CALL HIM
          MVC      PRM$RLEN-PARAMTER(2,R2),ROL$EXPL LARGE RECORD LENGTH
          LH       R14,ROL$EXPL   LARGE RECORD LENGTH
          ST       R14,PRM$TYPE+X'158' PLSYSUT1
          ST       R14,PRM$TYPE+X'15C' LSYSUT1
          L        R14,PRM$ADDR   SYSUT2'S RECORD ADDRESS
          LA       R15,yourstuf   YOUR SPECIAL EXPLOSION PARAMETER
          ICM      R15,B'1000',LASTLIST INDICATE LAST IN LIST

```

	STM	R14,R15,PRM\$WORK	EXPLODER PARAMETER LIST
*	CALL	exploder,PRM\$WORK	CALL YOUR EXPLOSION ROUTINE
	LA	R1,PRM\$WORK	+ADDRESS THE ADDRESS LIST
	L	R15,=V(exploder)	+<=== Your Explosion Program
	BALR	R14,R15	+CALL HIM
	MVC	PRM\$RLen-PARAMTER(2,R2),ROL\$EXPL	LARGE RECORD LENGTH
	B	ROL\$3900	BRANCH AROUND CONSTANTS
yourstuf	DC	C1120'parameter'	Special Parameter
ROL\$EXPL	DC	H'08000'	Exploded Record Length
.ROL3900	ANOP		
ROL\$3900	DS	0H	AROUND CONSTANTS

Figure 36. CPXIFACE Sample Code - Roll Your Own Slot

Potential Error Messages

- ROLL.YOUR.OWN - OPEN ERROR
- ROLL.YOUR.OWN - SYNCHRONOUS ERROR EXIT

Synchronizing Databases

There are at least two methods of synchronizing between databases. The SEGMENT keyword is designed for hierarchical structures such as DL/1. However, a variation of the KEY keyword called Random Key's is generally recommended in most cases. It becomes a matter of personal preference in choosing between the two.

The first step in synchronizing is to pair like records for comparison - ROOT is paired to ROOT, APPLES are paired to APPLES. Then, after like record types are paired, a control field is examined. In this way, if a record has been inserted or deleted, it can be identified as inserted or deleted.

Some records do not contain a control field. For example, some records add information to a database merely by their presence or by their relative position in a series. For such records, COMPAREX cannot be completely accurate in picking out the exact insertion or deletion, but the utility will show the series of differences, starting with the insertion or deletion.

Comparing

After the synchronizing process of pairing records (with either the SEGMENT or KEY keyword) IDENTITY, FIELD, and MASK keywords can be used to tell COMPAREX which bytes should be compared.

Refer to "Hierarchical Database Structure - Segment Synchrony" below for an example of comparing two versions (*unloaded or direct read*) of a database using SEGMENT synchronization logic.

Note: Random SEGMENTs are not recommended.

Segment Name	----- COMPAREX Keywords -----
ROOT	SEGMENT=(1 , EQ , C ' ROOT ' , (A , 09 , 5)) IDENTITY=(1 , EQ , C ' ROOT ') FIELD=(65 , END) MASK=(81 , 3)
APPLES	SEGMENT=(1 , EQ , C ' APPLES ') IDENTITY=(1 , EQ , C ' APPLES ') FIELD=(65 , END)
STEMS	SEG=(1 , EQ , C ' STEMS ' , (R , 23 , 4)) ID=(1 , EQ , C ' STEMS ') FIELD=(65 , END) MASK=(85 , 1) MASK=(93 , 1)

Figure 37. Hierarchical Database Structure - Segment Synchrony

Refer to "Database Structure - Random Key Synchrony" on page 53 for an example of comparing two versions (unloaded or direct read) of a database using a random KEY.

----- COMPAREX Keywords -----

```
KEY=(1,64,,R),BUFF=1024
  IDENTITY=(1,EQ,C'ROOT')
    FIELD=(65,END) MASK=(81,3)
  IDENTITY=(1,EQ,C'APPLES')
    FIELD=(65,END)
  ID=(1,EQ,C'STEMS')
    FIELD=(65,END),MASK=(85,1),MASK=(93,1)
```

Figure 38. Database Structure - Random Key Synchrony

ISPF Interface

The ISPF interface consists of:

- Panel library
- Messages library
- Clist library (*strictly for installation*)
- Sysgen library
- Object code library for module CPX\$ISPF

Processing Other Interfaces Disclaimer

The primary purpose of this ISPF interface is to simplify the procedure for specifying files and keywords for COMPAREX to use in its comparison process. If the files are QSAM, ISAM, VSAM, PDS, PANVALET, or LIBRARY, there is no problem. However, if you want to process exotic structures such as IDMS or DL/1 databases, the ISPF interface is not where you want to be. The reason for this is that these structures require so many specific allocations of system type libraries that it is not feasible for a generalized interface to be able to allocate all that are necessary.

Connected to Main Menu

Depending on how the interface was installed, you could go directly to "ISPF Panel CPX@PRIM" on page 55 or (*as recommended*) through "ISPF Panel VENDOR - Raw Form" below.

```
)BODY width(&zscreenw) expand(!!)
%!-! Vendor Supplied Software !-!
%Option ==>_ZCMD      +
%
% 1 +COMPAREX - Invoke%COMPAREX/ISPF
% 2 +CHANGE MAN - Invoke%Change Man+Services
% X +EXIT      -%Exit+to primary option menu
%
+Enter%END+command to return to primary option menu.
%
)PROC
  &ZSEL = TRANS( TRUNC (&ZCMD, '.')
    1, 'PANEL(CPX@PRIM) NEWAPPL(CPX) NEWPOOL'
    2, 'CMD(CMNINIT) NEWAPPL(CMN) NEWPOOL'
    , , ,
    X, 'EXIT'
    *, '?' )
)END
```

Figure 39. ISPF Panel VENDOR - Raw Form

The interface is panel driven. From your main menu, (ISR@PRIM), you eventually get to the COMPAREX primary menu - "ISPF Panel CPX@PRIM" below.

```
..... COMPAREX/ISPF 6.1.0 Primary Menu .....
Option ==>

0 - SHORT CUT - Single screen for options and dataset names
1 - OPTIONS   - Specify compare options for this session
2 - DSNAMES   - Specify dataset names to be compared
3 - SAVE      - Save options profile for future sessions
4 - LOAD      - Select/Delete options profile from prior session(s)
5 - CLEAR     - Clear previously loaded profile
6 - FOREGROUND - Invoke COMPAREX in the foreground and wait
7 - BACKGROUND - Invoke COMPAREX as a submitted batch job
8 - BACKGROUND - Similar to above but edit job (and optionally SUBMIT)
X - EXIT      - Exit

Press HELP KEY for tutorial assistance at any point;
Enter END Command to exit.
```

Figure 40. ISPF Panel CPX@PRIM

Note that the cursor is positioned at the

```
Option ==> _
```

where you may enter an option (0 to 8 or X) or TSO commands such as:

```
Option ==> TSO LISTC LEVEL(somnode.COMPAREX)
```

or

```
Option ==> TSO STATUS
```

TSO commands may be processed at any point (*except TUTORIAL*) within the COMPAREX interface. Speaking of TUTORIAL, it may be invoked at any point by pressing the Help Key (PFK1). The TUTORIAL will not be presented here, it is left as an exercise to the reader to walk through it.

Sample Sessions

For the discussion that follows, we will compare:

- DATA files - ABC.DATA.FILE1 versus ABC.DATA.FILE2
- TEXT files - ABC.TEXT.PDS(COB904AB) versus ABC.TEXT.PDS2(COB904AB)
- Directory-embedded files - SYS2.PANVALET.MASTER versus ABC.PDS

in these fashions:

- Option 6 - Foreground
- Option 0 - Short Cut
- Option 7 - Background Submit

- Option 3 - Background (edit before submit)

DATA Comparison in Foreground

Pick the options you want to use by entering 1 at:

Option ==> 1

Now, "ISPF Panel CPXOPDTO" below will be displayed asking you to make a decision between DATA, TEXT, and free-form keywords.

```
..... Data/Text Decision .....
Option ==> _

1 - DATA Comparison
2 - TEXT Comparison
3 - Enter free form keywords

NOTE - The options created in this particular session may be
       saved for future sessions by returning to the primary
       menu screen and using the SAVE option (3).

Press ENTER to continue; Enter END Command to exit.
```

Figure 41. ISPF Panel CPXOPDTO

DATA and TEXT are mutually exclusive. The third option - Enter free form keywords - gives us the option of specifying either DATA or TEXT and all of the associated keywords in native format.

Assume we want to make a **DATA** comparison, enter 1 at:

Option ==> 1

Now, "ISPF Panel CPXOPDAT" below will be displayed asking you to further clarify how you want to compare these DATA files.

```
----- Data Processing Options -----  
Option ==> _  
  
1 - Input processing      (FIELD MASK FILTER IDENTITY DESEN and Other)  
2 - KEY Synchronization  
3 - SEGMENT Synchronization  
4 - Display processing    (FORMAT MAXDIFF CASE PAGE LINE etc.)  
5 - Output processing     (COPYDIFF)  
6 - Enter free form keywords  
Press ENTER to continue; Enter END Command to exit.
```

Figure 42. ISPF Panel CPXOPDAT

All we know about these files is that there is a **KEY** in byte positions 6 through 19 (*relative to one*). Since we do not want a runaway compare at the first inserted record, we must specify some kind of Key Synchronization to isolate any inserted/deleted records, get back in synch and continue on.

We do this by entering 2 at:

```
Option ==> 2
```

Now, "ISPF Panel CPXOPSYK" below will be displayed allowing you specify the synchronizing KEY (up to five may be specified).

```

..... Data Synchronization by Key .....
Command ==> _

-----KEY ONE-----
DISPLACEMENT LENGTH  FORMAT  A/D/R
6      14      c      a
-----
-----
-----
-----

-----KEY TWO-----
DISPLACEMENT LENGTH  FORMAT
-----
-----
-----
-----

Press ENTER to register; Enter END Command to exit.

```

Figure 43. ISPF Panel CPXOPSYK

Since the KEY is in the same position in both files, we don't have to bother with the fields under

```

-----KEY TWO-----
DISPLACEMENT LENGTH  FORMAT

```

The displacement is 6 and the length is 14 (bytes 6 through 19 inclusive). The default format is C for character but the key itself could be combinations of packed, zoned, binary, and character data. The sequence is A for ascending and could have been left blank, because that is the default.

At this point, we should hit **ENTER** to register what we have specified³. After registering the specifications, issue the

- **END** **PFK3** command

or

- **RETURN** **PFK4** command.

If you repeatedly press **PFK3**, you will retrace your steps back through the panels just visited. That is OK. There is a quicker way, however. If you issue the **RETURN** command either explicitly or by its equated Program Function Key - **PFK4** perhaps - you will jump back to "ISPF Panel CPX@PRIM" on page 55 directly. Conversely, you could have "jumped" into "ISPF Panel CPXOPSYK" above by specifying

Option ==> 1.1.2

and then when registered, you could hit either **END** or **RETURN** which would bring you back to where you started from.

³ Otherwise, it is possible to pass dialogue variables containing garbage into module "CPXSISPF" which might abend later.

Assuming that we are back to "ISPF Panel CPX@PRIM" on page 55, we can either

- specify the dataset names now via Option 2 and then invoke COMPAREX

or

- invoke COMPAREX in the Foreground via Option 6 and be prompted for the dataset names.

Let's assume that no dataset names have been left over from any prior sessions⁴ and invoke COMPAREX in the foreground like this:

Option ==> 6

Now "ISPF Panel CPXOPDS1" below will be presented, requesting you to fill in a dataset name for SYSUT1.

```
..... File One (SYSUT1) .....
Command ==>

ISPF LIBRARY:
PROJECT ==> _____
LIBRARY ==> _____
TYPE ==> _____
MEMBER ==> _____ DATASET PASSWORD ==> _____ (If Protected)

OTHER PARTITIONED OR SEQUENTIAL DATASET:
DATASET NAME ==> _____
VOLUME SERIAL ==> _____ (If Not Cataloged)
UNIT ==> _____ (If Not Cataloged)

TYPE OF DATASET ==> _____ (COMPAREX Interface Only)
1 - PANVALET 2 - LIBRARIAN/GEM 3 - OTHER PROPRIETARY LIBRARY/DBMS
MEMBER NAME ==> _____ (16 Character Member Name)
INCLUDES ==> _____ (YES or NO - Expand Includes)
LEVEL ==> _____ (LIBRARIAN Only - Archie Level)
PARAMETER ==> _____ ('Parameter Data' for CPXIFACE)

Press ENTER to register; Enter END Command to exit.
```

Figure 44. ISPF Panel CPXOPDS1

Since the dataset name is three nodes in regular ISPF nomenclature, let's fill it in this way:

```
ISPF LIBRARY:
PROJECT ==> abc_____
LIBRARY ==> data_____
TYPE ==> file1_____
MEMBER ==> _____
```

and hit **ENTER**. Assuming that this dataset exists (*it is cataloged on an accessible disk pack*), the file for SYSUT1 is allocated.

⁴ Otherwise, we could have used Option 5 to clear out the dialogue variables first.

Now "ISPF Panel CPXOPDS2" below will be presented requesting you to fill in a dataset name for SYSUT2 in a similar fashion.

```

..... File Two (SYSUT2) .....
Command ==>

ISPF LIBRARY:
PROJECT ==> _____
LIBRARY ==> _____
TYPE ==> _____
MEMBER ==> _____ DATASET PASSWORD ==> _____ (If Protected)

OTHER PARTITIONED OR SEQUENTIAL DATASET:
DATASET NAME ==> _____
VOLUME SERIAL ==> _____ (If Not Cataloged)
UNIT ==> _____ (If Not Cataloged)

TYPE OF DATASET ==> _____ (COMPAREX Interface Only)
1 - PANVALET 2 - LIBRARIAN/GEM 3 - OTHER PROPRIETARY LIBRARY/DBMS
MEMBER NAME ==> _____ (16 Character Member Name)
INCLUDES ==> _____ (YES or NO - Expand Includes)
LEVEL ==> _____ (LIBRARIAN Only - Archie Level)
PARAMETER ==> _____ ('Parameter Data' for CPXIFACE)

Press ENTER to register; Enter END Command to exit.

```

Figure 45. ISPF Panel CPXOPDS2

This time we will enter the dataset name in the "Other" dataset name area and intentionally make an error this way:

```

OTHER PARTITIONED OR SEQUENTIAL DATASET:
DATASET NAME ==> 'abc.data.garbage' _____

```

and hit **ENTER**. Assuming that this dataset does not exist, we will receive a prompting message in the upper right hand corner of the panel:

```

... File Two (SYSUT2) ----- DSN not Cataloged

```

The dataset name we entered has been raised to uppercase and we may now overwrite part of it to correct it this way:

```

OTHER PARTITIONED OR SEQUENTIAL DATASET:
DATASET NAME ==> 'ABC.DATA.file2' _____

```

and hit **ENTER**. Assuming that this dataset does exist, we will now actually invoke COMPAREX in our region and await its completion.

In the meantime (*in terms of seconds usually*), we will have "ISPF Panel CPXRUNPR" below LOCK'ed on our screen.

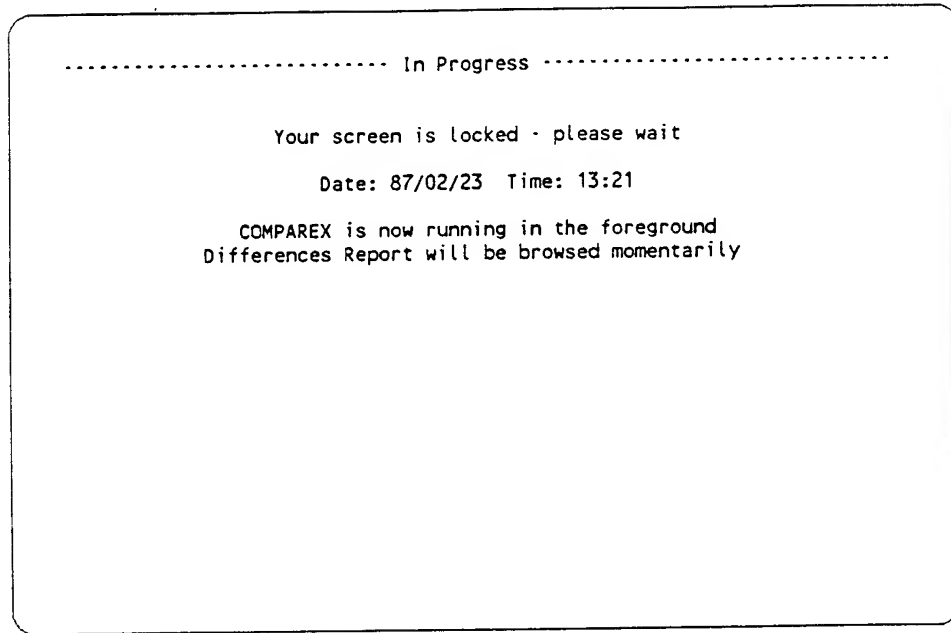


Figure 46. ISPF Panel CPXRUNPR

When COMPAREX is through comparing the two datasets, the Differences Report will be presented for browsing. At this point, we are under program (CPXSISPF) control - not panel control. Do not enter the RETURN command now. When you are finished scrolling⁵ through the Differences Report, hit **PFK3**.

⁵ Up and down, left and right.

Now "ISPF Panel CPXREPT" below will be presented requesting a disposition for the dynamically allocated SYSPRINT file.

```
----- Specify Report Disposition -----
Command ==>

REPORT DISPOSITION ==> K      (P=Print) (K=Keep) (D=Delete)
                                (PD=Print and delete)
                                (JD=Submit a job to print and delete)
                                (JK=Submit a job to print and keep)

LOCAL PRINTER ID
(Or Print Class)   ==> _____ (For Print option)

JOB STATEMENT FOR SUBMISSION IF REPORT DISPOSITION OPTION ABOVE IS JD OR JK

==> //yourida JOB (accounting info),MSGCLASS=A,CLASS=c,_____
==> //          NOTIFY=yourid_____
==> // *
==> // *

Press ENTER to continue; Enter END Command to exit.
```

Figure 47. ISPF Panel CPXREPT

The default is to **Keep** the dataset for further browsing at a later time. We will override that disposition and **Delete** it at this time to clean up after ourselves like this:

```
REPORT DISPOSITION ==> d  (P=Print) (K=Keep) (D=Delete)
```

Now hit **ENTER**. A message from TSO will penetrate the panel stating that

yourid.CPXyyddd.Thhmsst.OUTLIST

has been deleted and you are once again back to panel "ISPF Panel CPX@PRIM" on page 55.

TEXT Comparison Via Short Cut

Assuming we are at the COMPAREX/ISPF 6.1.0 Primary Menu again, choose the "Short Cut" method of specifying dataset names and brief options by entering 0 at:

```
Option ==> 0
```

Now, "ISPF Panel CPXSHORT" below will be displayed asking you to specify both file names and the free-form keywords required to compare them.

```
..... Dataset Names and Brief Options .....
Command ==>

SYSUT1 ISPF FILE:                SYSUT2 ISPF FILE:
PROJECT ==> _____          PROJECT ==> _____
LIBRARY ==> _____          LIBRARY ==> _____
TYPE   ==> _____          TYPE   ==> _____
MEMBER ==> _____          MEMBER ==> _____

OTHER PARTITIONED, SEQUENTIAL, VSAM, OR ISAM DATASET:
SYSUT1 DSNAME ==> _____
SYSUT2 DSNAME ==> _____

ENTER FREE FORM KEYWORDS BELOW: (No Syntax Checking Done on the Panel)
==> _____
==> _____
==> _____
==> _____

Press ENTER to register; Enter END Command to exit.
```

Figure 48. ISPF Panel CPXSHORT

We will enter the two dataset names to compare in the "ISPF FILE" nomenclature like this:

SYSUT1 ISPF FILE:	SYSUT2 ISPF FILE:
PROJECT ==> abc_____	PROJECT ==> abc_____
LIBRARY ==> text_____	LIBRARY ==> text_____
TYPE ==> pds_____	TYPE ==> pds2_____
MEMBER ==> cob904ab	MEMBER ==> cob904ab

Since we already know that these two members are Cobol program source code, we need only say:

```
ENTER FREE FORM KEYWORDS BELOW: (No Syntax Checking ...
==> text=cobol          /* Cobol text compare
==> maxdiff=100,continue /* Don't go nuts
==>
==>
```

and hit **ENTER**. After registering them to ISPF, hit **PFK3** to return to the main menu again. Now invoke COMPAREX as a submitted batch job like this:

```
Option ==> 7
```

Now "ISPF Panel CPXJOBST" below will be presented asking if there are any changes to be made to the **JOB** card image last used.

```
..... Job Statement .....
Command ==>

JOB STATEMENT:

====> //youridx JOB (accounting info),MSGCLASS=A,_____
====> //          CLASS=x,NOTIFY=yourid_____
====> //*_____
====> //*_____

Press ENTER to continue;
Enter END Command to terminate JOB submission.
```

Figure 49. ISPF Panel CPXJOBST

Now overwrite any changes to be made in the **JOB** card and hit **ENTER**. A message will penetrate the current panel⁶ like this:

```
JOB YOURIDX(JOB00345) SUBMITTED
```

This is the response to the TSO Submit command. There will be three asterisks "****" at the bottom of the screen. Hit **ENTER** one more time to bring us back to the primary menu. The background job we just submitted should be taking off shortly - depending on how loaded the machine is.

⁶ Dependent on how the ISPF interface was installed - TSO Submit or Internal Reader.

Compare Directories Via Edited Batch Job

Assuming we are at the COMPAREX/ISPF 6.1.0 Primary Menu again, choose different dataset names by entering 2 at:

Option ==> 2

Now, "ISPF Panel CPXOPDSN" below will be displayed asking you to specify one of SYSUT1 or SYSUT2.

```
..... Dataset Names .....
Option ==>

1 File one data set name      (SYSUT1)
2 File two data set name      (SYSUT2)
3 Output file data set name   (SYSUT3 - only used with COPYDIFF)

Press ENTER to continue; Enter END Command to exit.
```

Figure 50. ISPF Panel CPXOPDSN

We will choose SYSUT1 first by entering 1 at:

Option ==> 1

Now "ISPF Panel CPXOPDS1" on page 59 will again be presented requesting you to fill in a dataset name for SYSUT1. We will fill in the name of the PANVALET master file like this:

```
OTHER PARTITIONED OR SEQUENTIAL DATASET:
  DATASET NAME ==> 'sys2.panvalet.master'
  VOLUME SERIAL ==> _____ (If Not Cataloged)
  UNIT         ==> _____ (If Not Cataloged)

TYPE OF DATASET ==> 1 (COMPAREX Interface Only)
1 - PANVALET  2 - LIBRARIAN/GEM  3 - OTHER LIBRARY/DBMS
```

Now hit **ENTER**, correct any syntax errors, and hit **PFK3**. Again "ISPF Panel CPXOPDSN" above will be displayed asking you to pick SYSUT1, SYSUT2, or SYSUT3.

We will choose SYSUT2 by entering 2 at:

```
Option ==> 2
```

Now "ISPF Panel CPXOPDS2" on page 60 will again be presented requesting you to fill in a dataset name for SYSUT2. We will specify an existing partitioned dataset like this:

```
OTHER PARTITIONED OR SEQUENTIAL DATASET:
  DATASET NAME ==> 'abc.pds'
```

Now hit **ENTER**, correct any syntax errors, and hit **PFK4**. We are back to "ISPF Panel CPX@PRIM" on page 55. Construct a background job to be edited before submission like this:

```
Option ==> 8
```

Now we are editing (*via PDF EDIT*) a dynamically allocated job that can be optionally submitted at this point. Just after the JOB card images are some comments that you should pay attention to:

```
//*      C O M P A R E X    BATCH EXECUTION
//* YOU MAY EDIT THIS DATASET AT WILL.  WHEN YOU
//* ARE FINISHED, "SUBMIT" AT THE COMMAND ==> SUB
//* IF YOU DON'T SUBMIT, THE DATASET IS JUST DELETED
```

Since we didn't specify any options, or we got the ones carried over from last execution, we have the opportunity to modify them now. Find the SYSIN statement,

```
//SYSIN      DD *
```

There probably are a lot of commas following this statement. That is normal. They are always ignored by COMPAREX. Delete whatever you don't like and enter this:

```
DIR=PDF
```

We want to compare the directories only of the PANVALET master against the PDS. See "DIRECTORY" on page 83 for a discussion of what PDF means.

Now submit the job by keying in

```
COMMAND INPUT ==> sub
```

and hitting **ENTER**. The job will be submitted, a message will penetrate the current panel signifying the job number etc.

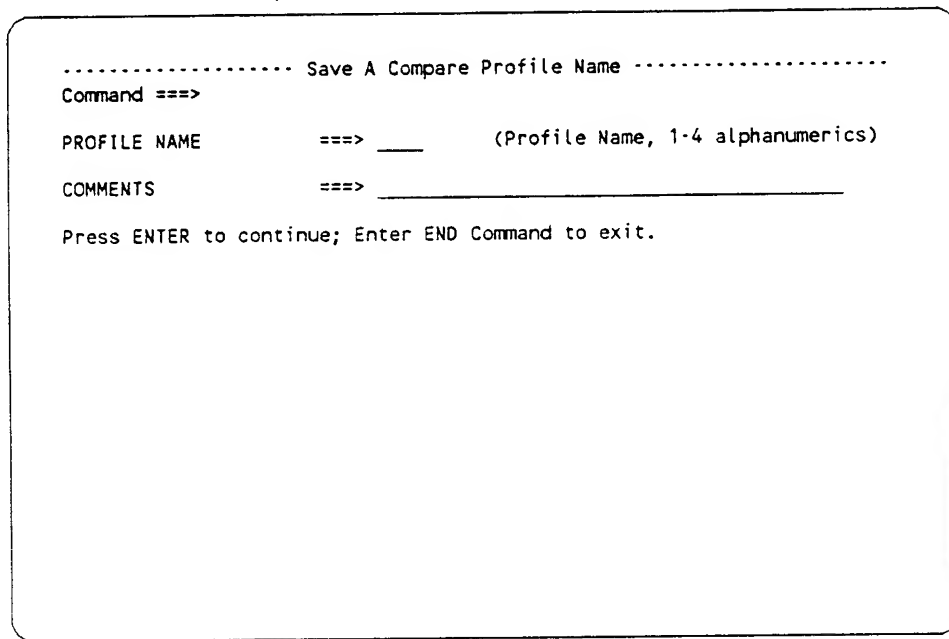
Now hit **PFK3** to end the edit session. The job dataset will automatically be deleted and we are again back to "ISPF Panel CPX@PRIM" on page 55.

Save and Load Profiles

At any point, we can decide that we have a good set of keywords for comparing files and we want to save it. From the primary menu, enter

Option ==> 3

Now, "ISPF Panel CPXPRSAV" below will be displayed asking you to name the profile. It must be one to four alphanumeric characters. Comments are strongly recommended.



```
..... Save A Compare Profile Name .....  
Command ==>  
  
PROFILE NAME      ==> ____ (Profile Name, 1-4 alphanumeric)  
  
COMMENTS          ==> _____  
  
Press ENTER to continue; Enter END Command to exit.
```

Figure 51. ISPF Panel CPXPRSAV

Give it a name and comments like this:

```
PROFILE NAME      ==> cob_ (Profile Name, 1-4 alphanumeric)  
  
COMMENTS          ==> Good Cobol comparison. _____
```

Now hit **ENTER**. You are automatically taken back to "ISPF Panel CPX@PRIM" on page 55 with the notification

COB saved

in the upper right corner

Option ==> 4

```
..... Select/Delete Existing Profile .....  
Command ==>  
  
SELECT    PROFILE      LAST-CHANGED  
S/D       NAME         DATE     TIME   ----- C O M M E N T S -----  
||||     COB          87/02/23  12:51   Good Cobol comparison.  
||||     DOUG         87/02/23  23:00   Csect parsing.  
||||  
||||  
||||  
||||  
||||  
||||  
||||  
||||  
||||  
||||  
||||  
||||  
||||  
||||  
||||
```

Press ENTER to continue; Enter END Command to exit.

We can Select only one, but we can Delete as many in the list as needed, like this:

Note: Deleting from this list does not free up the space in the profile pool. It merely opens up a spot in the internal directory record that also goes into the profile pool. If your profile pool fills up because you used too many of these profiles, you must either reuse some of the same names that have been deleted, or (*drastic measure*) delete and redefine your profile pool dataset.

Chapter 5 - Input Processing Keywords

The input processing routines set up the non-default parameters for the execution (based on the keywords), open the SYSUT1, SYSUT2, and SYSUT3 files, read the input file records, select records to send to the comparison routines, and pair records for comparison.

Input Processing Keywords

Table 1. Input Keyword Properties

Keyword	Function	Applicability		
		Data	Text	Directory
HELP=INPUT	Receive a 'canned' paragraph of cryptic keyword syntax and explanation on what it does.	✓	✓	✓
CONTINUE	Continue processing beyond the limit set in 'MAXDIFF' without printing. Statistics are kept for the portion not displayed.	✓	✓	✓
CPXIFACE	Exit module name for reading exotic file structures such as PAN-VALET, LIBRARIAN, DL/1, and ADABAS.	✓	✓	-
DATA	Data comparison logic for files of any record format and length that may contain binary, zoned, packed, or alpha characters.	✓	-	-
DESEN	Desensitize a portion of a record from both SYSUT1 and SYSUT2.	✓	-	-
DESEN1	Desensitize a portion of a record from SYSUT1.	✓	-	-
DESEN2	Desensitize a portion of a record from SYSUT2.	✓	-	-
DIRECTORY	Process the directory portion only of a directory-embedded dataset. Abbreviation DIR.	-	-	✓
FIELD	Comparison of a portion of a record in the same position on both SYSUT1 and SYSUT2.	✓	only 1	-
FIELD1	Similar to FIELD but specifies the displacement, length, and format of the portion on SYSUT1.	✓	only 1	-
FIELD2	Similar to FIELD but specifies the displacement, length, and format of the portion on SYSUT2.	✓	only 1	-
FILTERIN	Inclusive filter using 'AND' logic.	✓	✓	✓
FILTEROUT	Exclusive filter using 'AND' logic.	✓	✓	✓
FILTORIN	Inclusive filter using 'OR' logic.	✓	✓	✓
FILTOROUT	Exclusive filter using 'OR' logic.	✓	✓	✓
IDENTITY	Identifies a record type on SYSUT1 to allow processing of subsequent FIELD, MASK, and DESEN statements.	✓	-	-

continued ...

Keyword	Function	Applicability		
		Data	Text	Directory
MASK	Ignore comparison of a portion of a record in the same position on both SYSUT1 and SYSUT2.	✓	limited	-
MASK1	Similar to MASK but specifies the displacement, length, and format of the portion on SYSUT1.	✓	limited	-
MASK2	Similar to MASK but specifies the displacement, length, and format of the portion on SYSUT2.	✓	limited	-
MODE	User orientation. Applications specifies displacements relative to one. Systems specifies displacements relative to zero.	✓	✓	✓
SKIPUT1	Bypass the first set of records on SYSUT1.	✓	✓	-
SKIPUT2	Bypass the first set of records on SYSUT2.	✓	✓	-
STOPAFT	Maximum number of records to be read from SYSUT1.	✓	✓	✓
SYSUT1	Description of what file SYSUT1 looks like.	✓	✓	✓
SYSUT2	Description of what file SYSUT2 looks like.	✓	✓	✓
WILDCARD	Single character specification such that when used in Filters, Identity's, or Segments, any input value in that position tests true.	✓	✓	✓

SYSIN File Processed

COMPAREX opens SYSIN, reads until exhaustion, and examines each record for keywords.

Comments

If a SYSIN record has an asterisk in the first position, COMPAREX considers the entire record to be a comment, and it does not search for keywords on that record. COMPAREX prints the record on SYSPRINT, to the right of message number CPX00I. Additional comments may be placed to the right of legitimate keywords by starting them with a slash-asterisk "/*". Everything to the right of the slash-asterisk delimiter is considered comments. For example:

```
* This is a comment
MAXDIFF=10,CONTINUE /* This is also a comment */
```

HELP

If a SYSIN record contains the HELP keyword, COMPAREX prints the HELP canned response on SYSPRINT. See "HELP" on page 139 for more information about the HELP keyword.

Incorrect Keywords

COMPAREX examines each SYSIN record for correct and incorrect keywords. If COMPAREX finds an incorrect keyword or an incorrect parameter, the utility prints the SYSIN record on SYSPRINT, to the right of message number CPX00I, and COMPAREX underscores the incorrect characters and prints the literal "ERROR?" on the right.

Correct Keywords

If COMPAREX finds a correct keyword with its associated parameters, the utility prints the SYSIN record on SYSPRINT, to the right of message number CPX00I, but COMPAREX does not underscore any characters. It uses these

correct keywords to modify its default processing parameters.

Correct and Incorrect Keywords on Same SYSIN Record

On some SYSIN records, COMPAREX may find correct keywords as well as incorrect keywords. The incorrect keywords are underscored, and the correct keywords are used to modify COMPAREX default processing parameters.

End Of Data on SYSIN

When COMPAREX comes to the end of the SYSIN file, the utility issues messages CPX03I through CPX19I. These messages, described in detail in "Chapter 10 - Messages" on page 148, specify the processing parameters COMPAREX will use for this execution.

SYSUT1, SYSUT2, and SYSUT3 Opened

After SYSIN is processed and HALT = YES has not been specified, COMPAREX opens SYSUT1, SYSUT2, and possibly SYSUT3.

Note: If the user specifies PAN, LIB, or OTH within the SYSUT1 and/or SYSUT2 keywords, then the COMPAREX interface (CPXIFACE) takes care of opening, searching, reading, and closing those files. The DDNAME used is dependent on the file type and may be explicitly overridden by the DDNAME subkeyword of SYSUT1 and SYSUT2.

SYSUT1 and SYSUT2 Opened

COMPAREX determines dataset organization and dataset attributes from the JCL.

1. SYSUT1=DUMMY has not been specified or if the JCL does not point to a null file, COMPAREX opens SYSUT1. If the open is not successful, COMPAREX terminates with a condition code of 16. After file SYSUT1 has been successfully opened, COMPAREX issues message CPX21I to show the dataset name and the dataset attributes.
2. If SYSUT2=DUMMY has not been specified or if the JCL does not point to a null file, COMPAREX opens SYSUT2. If the open is not successful, COMPAREX processes as a print utility, printing onto SYSPRINT any SYSUT1 record that passes filtering tests. After file SYSUT2 has been successfully opened, COMPAREX issues message CPX22I to show the dataset name and the dataset attributes.
3. If SYSUT1=DUMMY and SYSUT2=DUMMY have both been specified, COMPAREX issues its end-of-processing messages and terminates.

SKIPUT1 and SKIPUT2

Next, COMPAREX skips over any input records, according to the values specified by the SKIPUT1 and SKIPUT2 keywords. If the SKIPUT1 parameter, as displayed with message CPX09I, is not zero, COMPAREX reads this number of records on SYSUT1. If the SKIPUT2 parameter is not zero, COMPAREX reads this number of records on SYSUT2. Then, COMPAREX displays message CPX26I to show how many records were skipped. If COMPAREX processes an end of file on either file while skipping records, COMPAREX issues message CPX71I to show the end of the file and continues to process.

SYSUT3 Opened

Then, if COPYDIFF was specified, the utility opens output file SYSUT3 and COMPAREX issues message CPX16I to show the dataset attributes.

SYSUT1 and SYSUT2 Read

COMPAREX reads SYSUT1 until the utility finds a record eligible for comparison. Then, COMPAREX reads SYSUT2 until it finds a record eligible for comparison.

STOPAFT

COMPAREX uses the value of the STOPAFT keyword to determine the maximum number of records to be read from either file. The default value is STOPAFT=999999999999, and no larger value can be specified.

CONTINUE

If MAXDIFF was specified and the number of differences specified by that keyword has been displayed on the Differences Report, COMPAREX prints message CPX67I and it writes no more input records on the Differences Report.

If CONTINUE was specified, the utility will print a second line with message CPX67I to say that COMPAREX will continue without printing.

If CONTINUE was not specified, the utility, as soon as it issues message CPX67I, issues its end-of-job counts and closes its files.

If CONTINUE was specified, COMPAREX continues to read files, select records for comparison, compare records, and write any SYSUT3 file. The end-of-job counters show the total input count and the total number of differences found.

If the user wants to know how many differences are found but has no need to see any records, the user could specify:

```
/* Job to See Statistics Only */
MAXDIFF=0,CONTINUE
```

DIRECTORY

The user may enter the DIRECTORY keyword to specify that COMPAREX read and compare only the directories of directory-embedded datasets such as PANVALET, PDS's, GEM, and LIBRARIAN. With DIRECTORY processing, the file portions of directory-embedded datasets are neither read nor compared.

Displacement

COMPAREX uses FIELD, FILTER, IDENTITY, MASK, KEY, and SEGMENT keywords to process input. These keywords contain displacement values. These displacement values tell COMPAREX where the data on each keyword starts in the record.

Some users like to think that the first position of any record is position 1; other users like to think that the first position of any record is position 0.

COMPAREX assumes, as its default MODE, that the first position of each record is position 1. Users who like to think that the first position of each record is position 0 may change the MODE by entering MODE=SYSTEMS. If the user enters the MODE=SYSTEMS keyword, COMPAREX will process all displacements on other keywords as if the first position of each record is position 0.

Selecting Records for Comparison

After reading a record, COMPAREX uses any filtering specifications to determine if a record is eligible for further processing.

The filtering keywords provide COMPAREX with a powerful facility to perform logical tests for record selection.

Keywords

The four filtering keywords are FILTERIN, FILTEROUT, FILTORIN, and FILTOROUT. Each filtering keyword has an abbreviated version:

FILTERIN	-	FilterIN	-	FIN
FILTEROUT	-	FilterOUT	-	FOUT
FILTORIN	-	FiltORIN	-	FORIN
FILTOROUT	-	FiltOROUT	-	FOROUT

The user may enter either the full spelling of the keyword or its abbreviation. In the discussion that follows, only the full spelling of the keyword will be used.

Form of Keywords

All filtering keywords take the form:

```
keywordname=( [ { MEMBER, } ] d1 [ -d2 ] , op , t 'vvvv' )
              [ { M, } ]
              [ { CSECT, } ]
```

where keywordname is FILTERIN, FILTEROUT, FILTORIN, or FILTOROUT.

If the FILTER is to include or exclude certain members of a directory-embedded dataset such as a PDS, the MEMBER or M option is used.

d1 is the displacement, relative to one, of the first position (left-most or high-order) of the field on which COMPAREX is to perform the logical test. If MODE=SYSTEMS is specified, the displacement is relative to zero. A range may be specified on the displacement by specifying a dash ("-") and a second displacement (d2) such that any value STARTING within this inclusive range satisfies (passes) the filter criteria.

op is the operation to be performed with the test. The values for op are:

```
LT - less than
LE - less than or equal
EQ - equal
NE - not equal
GE - greater than or equal
GT - greater than
```

t is the type of value that follows with 'vvvv.' t can be C for alphanumeric characters (each position of 'vvvv' represents one byte) or X for hexadecimal values (two positions of 'vvvv' represent one byte).

'vvvv' is the value to be tested. If the type is C, the value may be any character. If the type is X, the value may be composed of numeric values and the letters A through F. The WILDCARD value may be used in any positions of the value to be tested to indicate that any input data in those positions passes the FILTER test.

Inclusive Keywords

The two inclusive filtering keywords are FILTERIN and FILTORIN. The two inclusive keywords end in the letters 'IN'. These keywords direct COMPAREX to include the record that passes the test in further tests or in further processing.

"AND" Logic

The inclusive filter that uses "AND" logic is FILTERIN. "AND" logic is defined as one or more tests where the data being tested must pass all such tests to be eligible for further processing. COMPAREX converts the last FILTER keyword to an "AND" logic FILTER.

"OR" Logic

The inclusive filter that uses "OR" logic is FILTERIN. "OR" logic is defined as one or more tests where the data being tested must pass at least one test to be eligible for further processing.

Exclusive Keywords

The two exclusive filtering keywords are FILTEROUT and FILLTOROUT. The two exclusive keywords end in the letters 'OUT'. These keywords direct COMPAREX to exclude the record that passes the test from further processing.

"AND" Logic

The exclusive filter that uses "AND" logic is FILTEROUT. "AND" logic is defined as one or more tests where the data being tested must pass all such tests to be excluded from further processing.

"OR" Logic

The exclusive filter that uses "OR" logic is FILLTOROUT. "OR" logic is defined as one or more tests where the data being tested must pass at least one test to be excluded from further processing.

Pairing Records for Comparison

COMPAREX decides how to send records to the comparison routines based on the type of synchronization being done. In addition, COMPAREX uses IDENTITY, FIELD, and MASK keywords to pick out specific fields for comparison.

Types of DATA Synchronization

If TEXT has not been specified or if COMPAREX has nullified TEXT comparison because of inconsistencies, (see message CPX24A in "Chapter 10 - Messages" on page 148 to see which keywords nullify TEXT processing), COMPAREX uses DATA comparison logic to compare its input files.

The types of synchronization available for DATA comparison logic are:

- KEY Synchronization

If KEY's have been specified, or if SYSUT1 is either an ISAM or VSAM/KSDS file, KEY synchronization is used. This means that COMPAREX matches records by KEY. COMPAREX sends records to the comparison routines paired by KEY or, if a KEY from one file is unmatched on the other file, alone. The records sent alone are identified as key synchronization mismatches. For more information about KEY synchronization, see "Chapter 6 - Data File Synchronization Keywords" on page 99.

- SEGMENT Synchronization

If SEGMENT's have been specified, SEGMENT synchronization is used. This means that COMPAREX processes databases and matches segments by SEGMENT. COMPAREX sends records to the comparison routines paired by SEGMENT or, if a SEGMENT from one file is unmatched on the other file, alone. The records sent alone are identified as segmenting synchronization mismatches. For more information about SEGMENT synchronization, see "Chapter 6 - Data File Synchronization Keywords" on page 99.

- Same-Physical-Record-Number Synchronization

If neither KEY nor SEGMENT specifications have been made, the same-physical-record-number synchronization is used. COMPAREX sends each SYSUT1 record to the comparison routines paired to the same numbered record on SYSUT2 (that is, record number 1 on SYSUT1 is compared to record number 1 on SYSUT2 and record number 1001 on SYSUT1 is compared to record number 1001 on SYSUT2). If one file contains more records than the other file, the extra records from the longer file are sent to the comparison routines alone.

If FILTERs have been specified with same-physical- record-number synchronization, COMPAREX changes its procedure for sending records to the compare routines. If a record is FILTER'ed out, COMPAREX processes that record's file until the utility finds a record that is FILTER'ed in; then, it sends a pair of records to the compare routines.

- CSECT Parsing of load modules

Once invoked, CSECT Parsing logic extracts information about the load module from the ESD (External Symbol Dictionary) entries on the front of the load module and determines where each subsequent CSECT starts and ends. The IDR (IDentification Records) are ignored and only full text blocks are buffered.

Differences are underscored, CSECT name to CSECT name. Date and time stamps are underscored as well as uninitialized data areas (DS versus DC). CSECT's that are indeed different because of different source code will be dramatically different but CSECT's that are the same such as transient modules (*COBOL has a lot of ILBOxxx modules*) will show as identical.

TEXT

If TEXT processing is being done, the input processing routines do not pair records for comparison. Instead, the TEXT processing routines synchronize records based on record-to-record compares. For information about TEXT processing, see "Chapter 6 - Data File Synchronization Keywords" on page 99.

Comparing Only On Specific Fields

The user is able to direct that COMPAREX compare only certain bytes (instead of the entire record) by the use of the IDENTITY, FIELD, and MASK keywords.

Up to 400 IDENTITY, FIELD, MASK, and DESEN keywords may be used together in any COMPAREX run. Each FIELD1 and FIELD2 pair counts as one FIELD statement and each MASK1 and MASK2 pair counts as one MASK statement.

In addition, COMPAREX allows for a table of 2020 bytes to hold all IDENTITYs and DESENs. See the information about the calculation of the IDENTITY table space given later in this chapter, with the IDENTITY keyword information.

If COMPAREX has found any IDENTITY, FIELD (or FIELD1 and FIELD2 pair) MASK (or MASK1 and MASK2 pair) or DESEN (also DESEN1 and DESEN2) specifications, the utility will issue message CPX12I after SYSIN has been read. This message shows the IDENTITYs, FIELDS, MASKs, and DESENs that COMPAREX will use for the execution. If GENFLDS has been specified, this message gives the relative number for each FIELD on the GENFLDS printout.

FIELD

The user may specify that COMPAREX compare only certain positions of the records. When FIELDS are used, COMPAREX does not compare on the positions of the record that are not specified by FIELD keywords. When FIELDS and MASKs are not used, COMPAREX compares on all positions of the record.

Under TEXT processing, only one FIELD keyword may be specified. See "Chapter 7 - TEXT Keywords" on page 112 for more information about specifying a FIELD with TEXT.

If the positions to be compared are in the same places on both input records, the FIELD keyword is used. If the positions to be compared are in different displacements and/or the field lengths differ and/or the field types differ, then FIELD1 and FIELD2 pairs of keywords are used. For example:

```
      /* JOB to Compare on Certain Fields */
MAXDIFF=100,CONTINUE          /* Set MAXimum DIFFerences */
FIELD=(1,10)                  /* Compare bytes 1 thru 10 */
FIELD1=(20,7,P),FIELD2=(27,4,B) /* Compare Packed to Binary */
FIELD1=(40,8,Z),FIELD2=(31,5,P) /* Compare Zoned to Packed */
```

MASK

The user may specify that COMPAREX ignore certain positions of the records when comparing. When MASKs are used, COMPAREX does not compare on the positions of the records specified by the MASK keywords.

If the positions to be ignored are in the same places on both input records, the MASK keyword is used. If the positions to be ignored are in different places on the two input records, MASK1 and MASK2 pairs of keywords are used.

COMPAREX creates FIELDS from MASKs. This means that if the user entered only a MASK specifying that positions 17 through 20 were to be ignored, then

```
MASK=(17,4)
```

would generate:

```
FIELD=(1,16,C)
FIELD=(21,END)
```

If a MASK keyword generates only one FIELD, it may be used with TEXT processing.

IDENTITY

The user may specify that COMPAREX compare different fields on different record types. The IDENTITY keyword is used to specify a logical test, to be made on the record from file SYSUT1, to identify the record type, and the FIELD and MASK keywords that come after that IDENTITY keyword and before the next IDENTITY keyword are used to process any record that passes the IDENTITY test.

The WILDCARD value may be used in any position of the value to be tested to indicate that any input data in those positions passes the IDENTITY test.

Here is a simple example. The customer file has two record types; if position 8 is 'C', the record type is a customer header record and if position 8 is 'I', the record type is a customer invoice record. The keywords might look like this:

```
*   JOB to Compare Customer File
MAXDIFF=1000,CONTINUE /* Limit the Differences Report */
IDENTITY=(8,EQ,C'C')  /* Customer Header */
    FIELD=(1,3)         /* Customer data */
    FIELD=(13,47)       /* Customer name */
    FIELD=(72,END)      /* Rest of Customer record */
    MASK=(81,3)         /* Except time stamp */
IDENTITY=(8,EQ,C'I')  /* Customer Invoice */
    FIELD=(1,7)         /* Invoice number */
    FIELD=(22,8)        /* Invoice cross-reference */
    FIELD=(72,END)      /* Rest of Invoice record */
    MASK=(81,3)         /* Except time stamp */
```

The sample job would test the record from file SYSUT1 for the two record types, and if a customer header were found, COMPAREX would compare only on positions 1 through 3, 13 through 59, 72 through 80, and 84 through the end of the record. Then, if a customer invoice record were found, COMPAREX would compare only on positions 1 through 7, 22 through 29, 72 through 80, and 84 through the end of the record.

If some record were found that passed neither of the IDENTITY tests (a customer payment record, perhaps, with 'P' in position 8), COMPAREX would compare on all positions of that record.

If at least one IDENTITY is specified, COMPAREX generates a final IDENTITY with message CPX12I to show that all records that do not pass the user's IDENTITY tests will be handled in this way:

```
IDENTITY= (CATCH-ALL)  
FIELD= (1, END)
```

Under TEXT processing, no IDENTITY statement may be used.

When a record from file SYSUT2 is selected to be printed on the Differences Report, the sequence number from message CPX12I for any IDENTITY associated with the record is shown with the record, to the right of message number CPX52I.

Keywords for Input Processing

The input processing keywords are used to modify the default input processing routines.

CONTINUE

allows a continuation of processing beyond the limit set in MAXDIFF. The input files are read, the input processing keywords operate on the input, the files are compared according to the instructions in the data files synchronization keywords, any SYSUT3 file continues to be written, and records are counted for the end-of-job statistics line. The Differences Report shows only the number of differences specified by the MAXDIFF keyword, but the end-of-job statistics line shows the total number of records on the input files and the total number of differences.

Keyword Examples

```
CONTINUE
CONTINUE=NO
CONTINUE=(YES)
```

CPXIFACE

Keyword Format

```
CPXIFACE=xxxxxxxx
```

specifies the load module name for the COMPAREX interface. The source code for this exit module is provided. Unless otherwise specified by this keyword the default is CPXIFACE=CPXIFACE. See "Chapter 4 - Interfaces" on page 28 for a discussion of what interfaces have been developed so far.

Keyword Examples

```
CPXIFACE=CPXIFACE
CPXIFACE=(CPX$DLI)
CPXIFACE=CPXABEND
```

DATA

Keyword Format

```
DATA[=CSECT[ ,ADCON={ YES } ] ]  
                        { NO }
```

specifies that the files have an inter-record relationship and are not TEXT. A master file would be an example of a DATA file. DATA is the default. DATA file comparison logic can involve same physical record number synchronization (a record is compared to the same numbered record on the other input file), KEY, or SEGMENT synchronization (see the description in "Chapter 6 - Data File Synchronization Keywords" on page 99 for more information).

With DATA logic, COMPAREX can highlight differences at the byte or nibble (half-byte). In addition, the synchronization logic is more efficient than with TEXT processing unless Random KEYs (or SEGMENTs) have been specified.

COMPAREX can compare load modules (RECFM = U) in two ways:
as straight DATA

csect parsing (DATA = CSECT)

If a load module from one library is copied to another library and then the two load modules are compared, they are equal. However, if identical source code is compiled and linkage-edited into one library and the same process is done into another library before comparison, there will be differences.

In most cases, the differences will be in a few IDR records, any date/time stamp that the compiler puts in, and any uninitialized data areas. However, if the block sizes of the two libraries are different or the order of CSECT's linkage edited in are different, major differences are noted.

CSECT Parsing can be invoked by specifying:

DATA=CSECT

Once invoked, CSECT Parsing logic extracts information about the load module from the ESD (External Symbol Dictionary) entries on the front of the load module and determines where each subsequent CSECT starts and ends. The IDR (IDentification Records) are ignored and only full text blocks are buffered. The normal default BUFFER size is 60K for TEXT processing or DATA with Random KEYs. For CSECT Parsing, the BUFFER size is 256K. You may override it larger or smaller, but it is not recommended to lower this from 256K. If anything, it is recommended that the BUFFER go as large as possible - perhaps BUFF=1024.

Differences are underscored, CSECT name to CSECT name. Date and time stamps are underscored as well as uninitialized data areas (DS versus DC). CSECT's that are indeed different because of different source code will be dramatically different but CSECT's that are the same such as transient modules (*COBOL has a lot of ILBOxxx modules*) will show as identical.

If one Csect changed slightly, it sometimes happens that the V-type address (ADCON) constants are all that change in the rest of the Csects. You may or may not be interested in seeing the differences in the address constants. Now there is an alternative. You may specify:

DATA=(CSECT,ADCON=NO)

to have each address constant (A-type and V-type) nullified to binary zeroes before any comparison is done. The default is ADCON=YES, implying that address constants are to be left alone. ADCON=NO should result in a shorter difference report.

- If the load module has been linkage-edited with the Scatter attribute (PARM='SCTR'), COMPAREX cannot parse the CSECT's at all. This should not be a problem. Generally, this is only used in the MVS Nucleus.

Refer to "Sample Load Modules Composed of CSECTs" on page 80 for a graphic example of two load modules each with member name 'MEMBER1'.

Refer to "CSECT Parsing - Execution JCL and Keywords" on page 81 for a sample execution JCL and keywords to compare these two load modules.
Refer to "CSECT Parsing - Sample Difference Report" on page 81 for a sample difference report when comparing MEMBER1 in the two PDS's with these JCL and keywords.

SYSUT1-PDS1 (MEMBER1)

CSECT10 (identical)
CSECT20 (shorter)
CSECT30 (identical)
CSECT40 (inserted)

SYSUT2=PDS2 (MEMBER1)

CSECT10 (identical)
CSECT20 (longer)
CSECT25 (inserted)
CSECT30 (identical)

Figure 53. Sample Load Modules Composed of CSECTs


```

//COMPAREX EXEC PGM=COMPAREX,REGION=650K
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DSN=PDS1
//SYSUT2 DD DISP=SHR,DSN=PDS2
//SYSIN DD *
DATA=CSECT /* Invoke CSECT Parsing
BUFF=500 /* Get more than the 256K default
FORMAT=06 /* Interleave differing lines
MODE=SYSTEMS /* Displacements relative to zero
MAXDIFF=50,CONTINUE /* Generally advised */
FIN=(M,0,EQ,C'MEMBER1') /* Just compare 'MEMBER1'
/* EOJ

```

Figure 54. CSECT Parsing - Execution JCL and Keywords

```

CPX00I - DATA=CSECT /* Invoke CSECT Parsing
CPX00I - BUFF=500 /* Get more than the 256K default
CPX00I - FORMAT=06 /* Interleave differing lines
CPX00I - MODE=SYSTEMS /* Displacements relative to zero
CPX00I - MAXDIFF=50,CONTINUE /* Generally advised */
CPX00I - FIN=(M,0,EQ,C'MEMBER1') /* Just compare 'MEMBER1'
...
CPX25I - DATA=CSECT,BUFF=500,FORMAT=06,INTERLEAVE=1
...
C O M P A R E X (OS-6.1.0 - 87/054)
SYSUT1=PDS1(MEMBER1),SYSUT2=PDS2(MEMBER1)

CPX51I - CSECT=CSECT20
CPX52I - CSECT=CSECT20

00A0 4DF9F1F6 5DF4F4F1 60F7F2F3 F440 ... O N E
00A0 4DF9F1F6 5DF6F3F5 60F5F5F3 F540 ... T W O
      ..... -DIFFERENCE+
7760 00540200 08001900 ... O N E
7760 00540200 08001900 01180000 00007FFF ... T W O
      ++++++ ++++++ -DIFFERENCE+

CPX41I - CSECT=CSECT40 D I F O N E
CPX42I - CSECT=CSECT25 D I F T W O

```

Figure 55. CSECT Parsing - Sample Difference Report

DESEN

Keyword Format

DESEN=(ddd,t'vvvv')

specifies that COMPAREX is to desensitize (clobber) a portion of both records (SYSUT1 and SYSUT2). The displacement of the field is given in ddd and the length of the field is what it takes to contain t'vvvv'. For example, C'ABCD' is four bytes long while X'ABCD' is two bytes long.

If IDENTITYs are present, the desensitizer can be under it along with FIELDs and MASKs. The order of invocation is the order specified. If you want certain fields to be desensitized before comparing (and hence displaying the inequalities) then they must be specified chronologically ahead of the FIELDs.

For an example of how DESEN (DESEN1 and DESEN2 also) fit into the IDENTITY, FIELD, and MASK picture, see "IDENTITY, FIELD, MASK, and DESEN Messages" on page 91.

Keyword Examples

```
DESEN=(12,C'FORMER NAME FIELD  ' )
DESEN=(0042,X'0000000C')
```

DESEN1

Keyword Format

DESEN1=(ddd,t'vvvv')

specifies that COMPAREX is to desensitize (clobber) a portion of a record on SYSUT1. In general, it is used when displacements differ between like data in the two files. In all other respects, it is the same as DESEN.

DESEN2

DESEN2=(ddd,t'vvvv')

specifies that COMPAREX is to desensitize (clobber) a portion of a record on SYSUT2. In general, it is used when displacements differ between like data in the two files. In all other respects, it is the same as DESEN.

Keyword Examples

```
DESEN1=(12,C'FORMER NAME FIELD  ' )
DESEN2=(14,C'FORMER NAME FIELD  ' )
DESEN2=(0042,X'0000000C')
```

DIRECTORY

Keyword Format

```
DIRECTORY[={SPF} ]  
(or DIR) [ {PDF} ]  
          [ {USER} ]
```

specifies that COMPAREX is to read and compare only the directories of the following types of files in any combination:

Partitioned Datasets (PDS)

PANVALET masters through COMPAREX interface

LIBRARIAN masters through COMPAREX interface

GEM masters through COMPAREX interface

OTHER proprietary masters through COMPAREX interface

It is not necessary to specify an option to DIRECTORY. If no option is specified, COMPAREX processes only the member name. It will indicate an 'ALIAS' when appropriate, but no other data is used.

If DIRECTORY=USER is specified, COMPAREX will compare the user data in like named members. It is displayed in hexadecimal format on the Differences Report.

If DIRECTORY=SPF is specified, the user data is compared as in DIRECTORY=USER, but it is displayed on the Differences Report under special headings with proper formats. This happens only when partitioned datasets (PDS's) created and maintained by TSO/SPF or ISPF/PDF when STATS are on or when simulated by the COMPAREX interface in retrieving PANVALET, LIBRARIAN, or GEM members.

If DIRECTORY=PDF is specified, COMPAREX formats exactly as with DIRECTORY=SPF except that the dates (created and last modified) are presented in Gregorian order (YYMMDD) - not Julian. When IBM changes the PDF format in the future, this option will provide different headings and formats on the Differences Report.

When comparing two directory-embedded datasets, the size of the differences report can get rather hefty. Some of the ways to trim it down, unless of course you want to see it all, are to:

Filter (in or out) by member (generic) name.

Set MAXDIFF lower.

Set STOPAFT lower.

Sometimes a user is only interested in finding out what member names match between two libraries or conversely, find out what members do not match on member name. The PRINT keyword logic applies to member names for displaying.

In the following example, two libraries, LIB1 and LIB2 have as members:

LIB1	LIB2
A	A
B	
C	C
	D
E	E

If a DIRECTORY compare were made of these two libraries, the report would look something like this:

A		O N E 1
B	DIF	O N E 2
C		O N E 3
D	DIF	T W O 3
E		O N E 4

If PRINT=NOMATCH was specified, the report would look something like this:

B DIF O N E 2
D DIF T W O 3

If PRINT=NOMISMATCH was specified, the report would look something like this:

A	O	N	E	1
C	O	N	E	3
E	O	N	E	4

The sub-keyword options to DIRectory such as USER, SPF, and PDF would alter the display, particularly if there are differences in the directory data.

Keyword Examples

DIRECTORY=(PDF)
DIR=SPF
DIR

FIELD

Keyword Format

```
FIELD=(ddd,{111}[,C])
      {END}[,Z]
      [,P]
      [,B]
```

specifies comparison of a portion of a record. The field occurs in the same relative position on both input files. When any FIELD keyword is given, COMPAREX compares only those FIELDS (or FIELD1s and FIELD2s) given; it ignores the rest of the record during the comparison. Up to 400 IDENTITYs, FIELDS, MASKs, and DESENs may be used in one COMPAREX execution. END is used to signify through the end of the record. C stands for Character and any length may be specified; Z for Zoned and length cannot exceed 8; P for Packed and length cannot exceed 8; B for Binary and length cannot exceed 4.

It is not recommended that you specify Zoned, Packed, and Binary in FIELDS unless there is a very logical reason to do so. The reasons for this are:

1. If you specify Zoned, Packed, or Binary for a FIELD that indeed is not of that format, COMPAREX will abend when attempting to compare it to its counterpart in the other file. No precautions are made to check for format consistency in order to maintain speed.
2. When no exotic (Z, P, or B) specification is made, the default is Character. These comparisons (internally done as CLCL) are faster and just as accurate in under-scoring.

Keyword Examples

```
FIELD=(3,END)
FIELD=(12,4,P)
FIELD=(0999,8,Z)
FIELD=(15,00032,C)
```

FIELD1

Keyword Format

```
FIELD1=(ddd,{111}[,C])
      {END}[,Z]
      [,P]
      [,B]
```

specifies comparison of a portion of a record. The FIELD1 keyword is used to show the relative position, length, and format of the field on SYSUT1, and the FIELD2 keyword is used to show its counterpart on SYSUT2. The FIELD2 usually follows the FIELD1 keyword. When COMPAREX cannot find a paired FIELD2, it changes the FIELD1 to a FIELD.

In all other respects, FIELD1 is like FIELD.

Keyword Examples

```
FIELD1=(3,END)
FIELD1=(12,5,P)
```

FIELD2

Keyword Format

```
FIELD2=(ddd,[111][{,C}])  
           [{,Z}]  
           [{,P}]  
           [{,B}]
```

specifies comparison of a portion of a record. The field may differ in displacement, length, and format versus its associated FIELD1. See FIELD1 for further information about the FIELD2 keyword.

Keyword Examples

```
FIELD1=(7,END),FIELD2=9  
FIELD1=(12,7,P),FIELD2=(12,4,B)  
FIELD1=(22,8,Z),FIELD2=(23,5,P)
```

FILTERIN

Keyword Format

```
FILTERIN=( [ {MEMBER, } ] d1 [ -d2 ] , op , t 'vvvv' )  
(or FIN)  [ {M, }      ]  
          [ {CSECT, }   ]
```

specifies which record will be passed to the comparison routine or to the next FILTER test. If no FILTERs and no SKIPUTs are specified, all records are passed to the comparison routine.

d1 is the displacement, relative to one, of the first position (left-most or high-order) of the field on which COMPAREX is to perform the logical test. If MODE=SYSTEMS is specified, the displacement is relative to zero. A range may be specified on the displacement by specifying a dash ("-") and a second displacement (d2) such that any value STARTING within this inclusive range satisfies (passes) the filter criteria.

Values of 'op' are:

LT - less than

LE - less than or equal

EQ - equal

NE - not equal

GE - greater than or equal

GT - greater than

If the record passes the test specified by the FILTERIN, it is eligible for being tested by the next FILTER statement. If no further FILTER statements are specified, the record goes to the comparison routine. This FILTERIN keyword is an inclusive filter using 'AND' logic.

If the record fails any FILTERIN test, COMPAREX does not pass this record to the comparison routine. Instead, the utility reads another record from the input file.

The FILTERIN keyword tests records on both SYSUT1 and SYSUT2.

The WILDCARD value may be used in any positions of the value to be tested to indicate that any input data in those positions passes the FILTER test.

The user may specify that FILTERs include certain members of directory-embedded datasets such as PANVALET, LIBRARIAN, GEM, or PDS's by using the M or MEMBER option. The CSECT option is used only for CSECT Parsing (DATA=CSECT) such that certain CSECT names will be included in the comparison. Filtering of members, csects, and records may all be used in the same execution, however, it is not recommended that records be filtered in conjunction with CSECT Parsing.

COMPAREX provides a table area of 2020 bytes for filtering keywords. Each filtering keyword entered by the user takes up thirteen (13) bytes plus the length of the value (double if a wildcard is used). If the type of the value is character (C), each byte of the value between the apostrophes takes up one byte of the table; if the type of the value is hexadecimal (X), each byte of the value between the apostrophes takes up one-half byte of the table (an odd number of bytes is rounded up). For example:

```
FILTERIN=( 123 , EQ , X'00034 . ' )  
FILTERIN=( 45 , EQ , C'MASTERRECORD' )
```

would take up 44 of the table's 2020 bytes (13 for each of the two FILTERIN keywords, 6 for the six hexadecimal positions doubled for Wildcard, and 12 for the character value 'MASTERRECORD').

The M or MEMBER option for filtering keywords takes up no additional space in the FILTER table.

Keyword Examples

```
FILTERIN=( 07 , EQ , C'* ' )  
FILTERIN=( 3-60 , EQ , C'UNIT=3330 ' )
```

```

FILTERIN=(23,GT,X'01.....C')
FIN=(MEMBER,1,LE,C'CPX.A')
FIN=(CSECT,1,NE,C'ILB0.....')

```

FILTEROUT

Keyword Format

```

FILTEROUT=( [ {MEMBER, } ] d1 [-d2] , op , t 'vvvv' )
(or FOUT)  [ {M, }           ]
           [ {CSECT, }       ]

```

specifies which record will not be passed to the comparison routine. If the record passes the test specified by the FILTEROUT, and there are no more FILTER-type tests, COMPAREX does not pass this record to the comparison routine. Instead, COMPAREX reads another record from the input file.

If the record fails the test specified by the FILTEROUT, COMPAREX passes this record to the next FILTER-type test or to the comparison routine if no other tests are present.

In all other respects, it mechanically works the same as FILTERIN above.

Keyword Examples

```

FILTEROUT=(07,EQ,C' *')
FILTEROUT=(3-60,EQ,C'UNIT=3330')
FILTEROUT=(23,GT,X'01.....C')
FOUT=(M,1,LE,C'CPX.A')

```

FILTORIN

Keyword Format

```

FILTORIN=( [ {MEMBER, } ] d1 [-d2] , op , t 'vvvv' )
(or FORIN) [ {M, }           ]
           [ {CSECT, }       ]

```

specifies which record will be passed to the comparison routine. This FILTORIN keyword is an inclusive filter using 'OR' logic. As soon as the record passes this test, it is sent to the comparison routine. If the record does not pass this test, it is tested by the next FILTER-type test; if no other tests are present, the record is sent to the comparison routine.

Since COMPAREX converts the last FILTER keyword to an "AND" logic FILTER, a FILTORIN keyword will never be processed as the last FILTER keyword.

In all other respects, it mechanically works the same as FILTERIN above.

Keyword Examples

```

FILTORIN=(07,EQ,C' *')
FILTORIN=(3-60,EQ,C'UNIT=3330')
FILTORIN=(23,GT,X'01.....C')
FORIN=(M,1,LE,C'CPX.A')
FORIN=(CSECT,1,GE,C'PQA')

```


FILTOROUT

```
FILTOROUT=( [ {MEMBER, } ] d1 [-d2] , op, t'vvvv' )  
(or FOROUT) [ {M, } ]  
               [ {CSECT, } ]
```

specifies which record will not be passed to the comparison routine. If the record passes the test specified by the FILTOROUT, COMPAREX does not pass this record to the comparison routine. Instead, COMPAREX reads another record from the input file. If the record fails the test specified by the FILTOROUT, COMPAREX passes this record to the next FILTER-type test or to the comparison routine if no other tests are present.
In all other respects, it mechanically works the same as FILTEROUT above.

IDENTITY

Keyword Format

IDENTITY=(ddd,op,t'vvvv')
(or ID)

identifies a record on SYSUT1 so that the FIELD and MASK keywords that follow (until the next IDENTITY) can be used for this record.

Values of op are:

LT - less than

LE - less than or equal

EQ - equal

NE - not equal

GE - greater than or equal

GT - greater than

If the record on SYSUT1 does not pass any IDENTITY test, COMPAREX will compare all positions of the SYSUT1 record to the matched SYSUT2 record.

The WILDCARD value may be used in any positions of the value to be tested to indicate that any input data in those positions passes the IDENTITY test.

COMPAREX provides a table area of 2020 bytes for IDENTITY and DESEN keywords. Each IDENTITY keyword entered by the user takes up thirteen (13) bytes plus the length of the value (double if a Wildcard is used). If the between the apostrophes takes up one byte of the table; if the type of the value is hexadecimal (X), each byte of the value between the apostrophes takes up one-half byte of the table (an odd number of bytes is rounded up). For example:

IDENTITY=(123,GE,X'00034A')

ID=(45,EQ,C'MASTERRECORD')

would take up 41 of the table's 2020 bytes (13 for each of the two IDENTITY keywords, 3 for half of the six hexadecimal positions, and 12 for the character value 'MASTERRECORD').

Keyword Examples

IDENTITY=(07,EQ,C'A.4')

ID=(49,GT,X'07450F')

```

CPX031 - EXECUTION OF OS$JOB.CPX.SC1212 - VALUES EXTRACTED/DEFAULTED:
CPX041 - MAXDIFF=3,CONTINUE,STOPAFT=1000
CPX051 - PRINT=(MATCH,MISMATCH),MBRHDR=YES,HALT=COND,KEYSONLY
CPX061 - WILDCARD=C'.' ,MODE=APPLICATIONS, (ALL DISPLACEMENTS RELATIVE TO ONE)
CPX071 - SYNCHRONIZATION KEY(S):
        KEY1=(1,8,Z,R),KEY2=(2,7,Z)
CPX081 - DECIMAL,EBCDIC,CASE=UPPER,LINE=(32,HORIZONTAL),PAGE=58
CPX111 - DASH=C'-' ,PLUS=C'+' ,FLDSONLY
CPX121 - IDENTITIES, DESENSITIZING, FIELDS, AND MASKS:
        IDENTITY=(7,LE,C'3')
        DESEN=(33,C'ID LE 3')
        FIELD1=(5,1,Z),FIELD2=(6,2,Z)
        MASK=(15,8,Z)
        DESEN=(43,C'ID LE 3')
        IDENTITY=(7,EQ,C'5')
        MASK=(5,90,C)

        IDENTITY=(7,GE,C'6')
        DESEN1=(65,C'ID GE 6')
        DESEN2=(97,X'ABCDEF123.')
        FIELD=(30,80,C)
        FIELD1=(9,8,P),FIELD2=(17,4,8)
        FIELD1=(17,4,B),FIELD2=(21,4,8)
        IDENTITY=(CATCH-ALL)
        FIELD=(1,END)

        1 IDENTITY=(7,LE,C'3')
        2 DESEN=(33,C'ID LE 3')
        3 FIELD1=(5,1,Z),FIELD2=(6,2,Z)
        4 DESEN=(43,C'ID LE 3')

        5 IDENTITY=(7,EQ,C'5')
        6 FIELD=(1,4,C)
        7 FIELD=(95,END)
        8 IDENTITY=(7,GE,C'6')
        9 DESEN1=(65,C'ID GE 6')
        10 DESEN2=(97,X'ABCDEF123.')
        11 FIELD=(30,80,C)
        12 FIELD1=(9,8,P),FIELD2=(17,4,8)
        13 FIELD1=(17,4,B),FIELD2=(21,4,8)
        14 IDENTITY=(CATCH-ALL)
        15 FIELD=(1,END)

```

Figure 56. *IDENTITY, FIELD, MASK, and DESEN Messages*

IGNORSIN

when comparing DATA files and the basic difference between them is that one file contains packed fields with a sign of 'C' and the other contains packed fields with a sign of 'F', every record contains differences.

This keyword causes COMPAREX to scan every byte of both synchronized records for packed fields and make them signs of 'F' before comparison begins. When the two records are compared, these sign differences are effectively ignored.

Certain restrictions apply to the use of IGNORSIN:

No FIELD's or MASK's are allowed.

Crippled for TEXT processing.

Not applicable for CSECT Parsing

Keyword Examples

```

IGNORSIN
IGNORSIN=YES

```

MASK

Keyword Format

```
MASK=(ddd,{111}[,C])
      {END}[,Z]
      [,P]
      [,B]
```

specifies ignoring a portion of a record in the same relative position on both input files. This area in the records is not eligible for comparison.

When an IDENTITY keyword precedes the MASK keyword, the MASK keyword is used only when the record passes the IDENTITY test. When an IDENTITY keyword does not precede the MASK keyword, the MASK keyword applies to all records on the files.

END is used to signify through the end of the record. C stands for Character and any length may be specified; Z for Zoned and length cannot exceed 8; P for Packed and length cannot exceed 8; B for Binary and length cannot exceed 4. If Z, P, or B is specified for format, it is only cosmetic.

Keyword Examples

```
MASK=(256,END)
MASK=(83,5,Z)
MASK=(83,5)
```

MASK1

Keyword Format

```
MASK1=(ddd,{111}[,C])
      {END}[,Z]
      [,P]
      [,B]
```

specifies ignoring a portion of a record. The MASK1 keyword is used to show the relative position, length, and format of the field on SYSUT1, and the MASK2 keyword is used to show its counterpart on SYSUT2. The MASK2 usually follows the MASK1 keyword. When COMPAREX cannot find a paired MASK2, it changes the MASK1 to a MASK.

In all other respects, MASK1 is like MASK.

Note: MASK1 and MASK2 only have applicability if they fall under a FIELD1/FIELD2 pair. By themselves, MASK1/MASK2 will not generate (compile into) the proper FIELD1/FIELD2 statements.

Keyword Examples

```
MASK1=(256,END)
MASK1=(83,5,P)
MASK1=(83,5)
```

MASK2

Keyword Format

```
MASK2=(ddd,[111][{,C}])  
          [{,Z}]  
          [{,P}]  
          [{,B}]
```

specifies ignoring a portion of a record. The MASK2 keyword is used to show the relative position, length, and format of the field on SYSUT2, and the MASK1 keyword is used to show its counterpart on SYSUT1. The MASK2 usually follows the MASK1 keyword.

Keyword Examples

```
MASK2=(83,5,C)  
MASK2=(83,5)
```

MODE

Keyword Format

```
MODE={ APPLICATIONS }  
      { APL }  
      { SYSTEMS }  
      { SYS }
```

specifies the user orientation. MODE=APPLICATIONS or MODE=APL is the default. Under the applications mode, all displacements given on other keywords are relative to one. The first position of the record is position one.

In addition, under the applications mode, the LLBB (or RDW) of a variable length record is not processed. This means that the LLBB is not counted when determining the first position of the record, the LLBB is not shown on the Differences Report, the LLBB is not compared, and the LLBB is not able to be accessed with FIELD, FILTER, IDENTITY, MASK, and KEY statements.

Under the systems mode (MODE=SYSTEMS or MODE=SYS), all displacements on other keywords are relative to zero. The first position of the record is position zero.

In addition, under the systems mode, the LLBB (or RDW) of a variable length record is able to be processed. This means that the LLBB is counted when determining the first position of the record, the LLBB is shown on the Differences Report, the LLBB is compared, and the LLBB is able to be accessed with FIELD, FILTER, IDENTITY, MASK, and KEY statements.

Finally, under the systems mode, HEX becomes the default. The user may override this by entering a DECIMAL keyword along with the MODE=SYSTEMS keyword.

Keyword Examples

```
MODE=APL  
MODE=(SYSTEMS)
```

SKIPUT1

Keyword Format

```
SKIPUT1=nn
```

allows COMPAREX to skip over any desired number of records at the beginning of file SYSUT1 before comparison or printing begins.

If directory-embedded datasets are being read, COMPAREX will skip over that number of records at the beginning of each member.

Keyword Examples

```
SKIPUT1=00001  
SKIPUT1=500
```

SKIPUT2

Keyword Format

```
SKIPUT2=nn
```

allows COMPAREX to skip over any desired number of records at the beginning of file SYSUT2 before comparison or printing begins.

If directory-embedded datasets are being read, COMPAREX will skip over that number of records at the beginning of each member.

Keyword Examples

```
SKIPUT2=00001  
SKIPUT2=500
```

STOPAFT

Keyword Format

STOPAFT=nn

(default is 99999999999) specifies the maximum number of records to be read from SYSUT1 or SYSUT2. This number does not include records bypassed as a result of SKIPUT1 or SKIPUT2. COMPAREX stops processing as soon as the number is reached on either file, writes the statistics line, and closes all files.

For example, if STOPAFT=60 were specified and there were 50 records on file SYSUT1 and 70 records on file SYSUT2, COMPAREX would read all 50 records on file SYSUT1 and 60 records on file SYSUT2 before stopping its processing (with return code 8).

Keyword Examples

```
STOPAFT=00001  
STOPAFT=500
```

SYSUT1

Keyword Format

```
SYSUT1=(PAN) [, (MEMBER=)xxx] [, INCLUDE=(NO )] [, DDNAME=xxx] [, LEVEL=n] [, PARM='xxx'] )  
      (LIB) [, (M=)      ] [YES]  
      (OTH)
```

specifies the parameters to be passed to the COMPAREX interface (CPXIFACE) to read PANVALET, LIBRARIAN, GEM, and OTHER proprietary file structures in the place of SYSUT1.

PAN - PANSOPHIC's PANVALET library.

LIB - Applied Data Research's LIBRARIAN library or FUJITSU/FACOM's GEM, depending on how CPXIFACE has been generated.

OTH - any other proprietary library management package or database management system. The source code to CPXIFACE is supplied with PAN and LIB usable immediately. For the list of library/database management interfaces available in "OTH", see "Chapter 4 - Interfaces" on page 28.

MEMBER - a particular member of the library has been requested. **MEMBER** may be abbreviated M. The member name may be up to sixteen (16) alphanumeric characters. The absence of the **MEMBER** option implies that the entire library is to be processed.

INCLUDE - specifies whether or not included members (++**INCLUDE** in PAN and -**INC** in LIB) are to be expanded when read. The default is **NO**. If an option to **COPY-DIFF** is specified, any **YES** option to **INCLUDE** will be nullified to **INCLUDE=NO**.

DDNAME - specifies what ddname is to be used to open the proprietary file structure. For disk libraries, the default for PAN is PANDD1 and the default for LIB is MASTER. Certain other specifications mean special handling within CPXIFACE depending on the library. Unique ddnames and their meanings are:

PAN

PANDD1 - disk master

BACKUP - protection file

PANDD3 - tape master

PANDD4 - disk protection

LIB

MASTER - disk master

MASTIN - tape master

CYCLE - cycle control

LEVEL - (for LIB only) specifies the relative level number of the archived (ARCHIE) module. **LEVEL**=0 is the current module. **LEVEL**=1 is the next oldest and so on.

PARM - specifies extra data to be passed as is into CPXIFACE.

```
SYSUT1=(OTH,PARM='information for CPXIFACE')
```

LIBrarian can also use the **PARM** to take a dynamic 'Current Management Code.' Instead of a statically generated CMC, it can be passed into the (CPXIFACE) Interface to read 'PROD2' modules:

```
SYSUT1=(LIB,PARM=1234,MEMBER=abc)
```

Keyword Examples

```
SYSUT1=(PAN, MEMBER=PANMEMBER)
```

```
SYSUT1=LIB
```

```
SYSUT1=(OTH, M=MEMBER-IS-16-BYT, DDNAME=A, PARM='A B')
```

```
SYSUT1=(LIB, M=ABC, INCLUDE=YES, LEVEL=3, PARM=1234)
```


Alternate Keyword Format

```
SYSUT1={ QSAM }  
        { ISAM }  
        { [ ( [ VSAM ] , PASSWORD=password ) ] }  
        { PDS }  
        { DUMMY }
```

overrides the dataset organization (DSORG) determined by COMPAREX for SYSUT1. This is allowed but not recommended.

SYSUT2

Keyword Format

```
SYSUT2=( (PAN) [ , (MEMBER=)xxx ] [ , INCLUDE=(NO) ] [ , DDNAME=xxx ] [ , LEVEL=n ] [ , PARM='xxx' ] )  
        (LIB) [ , (M=) ] [ (YES) ]  
        (OTH)
```

specifies the parameters to be passed to the COMPAREX interface (CPXIFACE) to read PAN, LIB, and OTH proprietary file structures in the place of SYSUT2. It is identical in syntax to keyword SYSUT1.

Keyword Examples

```
SYSUT2=( PAN , MEMBER=PANMEMBER )  
SYSUT2=LIB  
SYSUT2=( OTH , M=MEMBER-IS-16-BYT , DDNAME=A , PARM='A B' )  
SYSUT2=( LIB , M=ABC , INCLUDE=YES , LEVEL=3 )
```

Alternate Keyword Format

```
SYSUT2={ QSAM }  
        { ISAM }  
        { [ ( [ VSAM ] , PASSWORD=password ) ] }  
        { PDS }  
        { DUMMY }
```

overrides the dataset organization (DSORG) determined by COMPAREX for SYSUT2. This is allowed but not recommended.

WILDCARD

Keyword Format

```
WILDCARD={ C'.' }  
          { t'vv' }
```

specifies the character used in any subsequent logical test to indicate that any data value passes the test.

The logical tests occur in FILTER, IDENTITY, and SEGMENT keywords.

Any number of WILDCARD keywords may be entered. The WILDCARD value (or default) is in effect until another WILDCARD is encountered.

For example, the keywords could be:

```
FILTERIN=( 12, EQ, X'.0.2' )  
WILDCARD=C'?'  
FILTERIN=( 23, EQ, C'AB??C' ) , TEXT=?  
WILDCARD=C'!'  
FILTERIN=( 34, EQ, C'DE!F!G' )
```

and COMPAREX would interpret the periods in the first FILTERIN; the question marks in the second FILTERIN and TEXT; the exclamation points in the third FILTERIN; all as WILDCARD values.

If the default (a period) were in effect and the user entered

```
FILTERIN=( 123, EQ, X'...D' )
```

then any data value with 'D' in the low-order half-byte would pass this test for negative numbers.

Any number of the positions of the value to be tested may contain the WILDCARD character. For example:

```
IDENTITY=( 234, EQ, C'...A...B...C...1' )  
SEGMENT=( 345, EQ, X'F....0' )
```

Keyword Examples

```
WILDCARD=C'*'  
WILDCARD=X'5C'
```

Chapter 6 - Data File Synchronization Keywords

COMPAREX recognizes two categories of files - DATA and TEXT - and the utility has a comparison logic routine for each. This chapter discusses DATA files and the DATA comparison logic routines; "Chapter 7 - TEXT Keywords" on page 112 discusses TEXT files and the TEXT comparison logic routines.

When TEXT has not been specified or when it has been nullified because of inconsistencies, COMPAREX uses its DATA comparison logic routines to compare the two input files.

When COMPAREX performs DATA comparison logic, it cannot also perform TEXT comparison logic in the same run. For this reason, the DATA keyword and the TEXT keyword are mutually exclusive; they may not both be entered by the user in the same execution. If both keywords are entered in the same execution, COMPAREX will use the last one on the file. DATA is the default.

Data File Synchronization Keywords

Table 2. Data File Synchronization Properties

Keyword	Function	Applicability		
		Data	Text	Directory
HELP = DATA	Receive a 'canned' paragraph of cryptic keyword syntax and explanation on what it does.	✓	-	-
KEY	Control field used to determine out-of-synchronization records. Maximum of 40.	✓	-	-
KEY1	Similar to KEY but specifies the displacement, length, and format of the portion on SYSUT1.	✓	-	-
KEY2	Similar to KEY but specifies the displacement, length, and format of the portion on SYSUT2.	✓	-	-
SEGMENT	Control field used to determine out-of-synchronization segments in databases. Recommend use of a single Random KEY.	✓	-	-

What is DATA?

DATA is defined, in COMPAREX, as any file where there is a known inter-record relationship. DATA files have bytes and fields in fixed relationships on their records.

Examples of DATA files are system master files, system intermediate files, system transaction files, load modules, and databases.

What is DATA Comparison Logic?

Under DATA comparison logic, the COMPAREX input processing routines attempt to match records, one record from each input file, to send to the compare routines. The three types of matching are KEY synchronization, SEGMENT synchronization, and physical-record to physical-record (no synchronization).

DUMMY Files

If the user specifies SYSUT1=DUMMY or SYSUT2=DUMMY, these DATA compare routines are not used. Instead, the input processing routines send the records from the non-DUMMY file directly to the Differences Report and records from SYSUT2 are sent to the output processing routines if COPYDIFF has been specified.

KEY Synchronization

If the user specified one or more KEY keywords, COMPAREX uses KEY synchronization to pair records to send to the compare routines. In addition, if no KEY was specified and the SYSUT1 file organization uses a KEY (such as ISAM or VSAM-KSDS), COMPAREX will use the KEY for synchronization.

KEYs are file control fields. Up to 40 KEYs (or KEY1 and KEY2 pairs) may be specified in each COMPAREX run. The first KEY keyword specified is the most major KEY on the file and the last KEY keyword specified is the most minor KEY on the file.

An example of the use of KEYs would be for a customer file. The most major KEY would be for the customer number field, an intermediate KEY would be for invoice number, and the most minor KEY would be for transaction date/time stamp. The KEY keywords could look like this:

```
KEY=( 3 , 7 )  
KEY=( 11 , 4 , P , D )  
KEY=( 31 , 6 , , R )
```

The KEY synchronization processing is as follows:

File Out Of Sequence

If, while reading records from either SYSUT1 or SYSUT2, COMPAREX finds that a KEY or SEGMENT control field is not in the sequence that has been specified (Ascending or Descending), then message CPX36A will be issued once followed by the offending record. For example, if a file consists of records with KEYs 1, 5, 10, 3, and 20; the record with KEY 3 is out of synch (Ascending assumed) with the rest. It would be noted with message CPX36A and printed, but any future "out-of-synch" situations would not be flagged - only the first violation.

If it is known in advance that DATA records are not in any pure ascending or descending sequence, then Random KEYs should be considered.

Duplicate KEY on the Same File

If the compare routines find that a set of KEYs on a record is equal to the set of KEYs on the previous record from that file, COMPAREX attempts to match this record to a record from the other file. Equal KEYs, then, cause no problems with synchronization, except that if only one record for that KEY exists on the other file, it will be paired for comparison to the first record on the file with the duplicate KEY and the second record on the file with the duplicate KEY will be identified as a KEY synchronization mismatch, showing on the Differences Report after message CPX61I (for records from file SYSUT1) or CPX62I (for records from file SYSUT2).

KEY Goes Beyond Record

If the starting position of any KEY goes beyond the end of the record, COMPAREX will terminate immediately, issuing message CPX35A.

End Of Data on SYSUT1

If file SYSUT1 has come to end of file and file SYSUT2 has not come to end of file, COMPAREX sends all extra records from file SYSUT2 to the Differences Report, showing them as extra records with message CPX62I. If COPYDIFF is in effect, COMPAREX sends the SYSUT2 records to the output processing routines for writing to file SYSUT3.

End Of Data on SYSUT2

If file SYSUT2 has come to end of file and file SYSUT1 has not come to end of file, COMPAREX sends all extra records from file SYSUT1 to the Differences Report, showing them as extra records with message CPX61I.

Records Matched on KEY

If the compare routines find that a set of KEYs on a record from file SYSUT1 is equal to a set of KEYs on a record from file SYSUT2 (the pair has been sent together to the compare routines synchronized by KEY), COMPAREX compares the data values in the records.

Records are Equal

If all bytes in the two records compare equal, COMPAREX bypasses the records. The utility does not send the records to the Differences Report, and it does not send the SYSUT2 record to the output processing routines for writing to file SYSUT3.

Using FIELDS

If FIELDS have been specified or if FIELDS have been compiled from MASK statements, COMPAREX compares just the bytes defined by FIELD keywords.

FIELDS Compare Equal

If all bytes in these FIELDS compare equal, COMPAREX bypasses the records. The utility does not send the records to the Differences Report, and it does not send the SYSUT2 record to the output processing routines for writing to file SYSUT3.

Records Compare Unequal

If all bytes in these FIELDS do not compare equal or if no FIELDS have been specified and the records do not compare equal or if no FIELDS have been specified and one record is longer than the other record, COMPAREX sends both records to the Differences Report, showing the SYSUT1 record with message CPX51I and showing the SYSUT2 record with message CPX52I. If COPYDIFF is in effect, COMPAREX sends the SYSUT2 record to the output processing routines for writing to file SYSUT3.

No Match on KEY for SYSUT1

If the KEY is an ascending KEY (the default) and the compare routines find a set of KEYs on a record from file SYSUT1 that is lower than the next set of KEYs on a record from file SYSUT2, COMPAREX sends the SYSUT1 record to the Differences Report, showing it as a KEY synchronization mismatch with message CPX61I.

If the KEY is a descending KEY (shown with the „D option on the KEY or KEY1 keyword) and the compare routines find a set of KEYs on a record from file SYSUT1 that is higher than the next set of KEYs on a record from file SYSUT2, COMPAREX sends the SYSUT1 record to the Differences Report, showing it as a KEY synchronization mismatch with message CPX61I.

No Match On KEY for SYSUT2

If the KEY is an ascending KEY (the default) and the compare routines find a set of KEYs on a record from file SYSUT2 that is lower than the next set of KEYs on a record from file SYSUT1, COMPAREX sends the SYSUT2 record to the Differences Report, showing it as a KEY synchronization mismatch with message CPX62I. If COPYDIFF is in effect, COMPAREX sends the SYSUT2 record to the output processing routines for writing to file SYSUT3.

If the KEY is a descending KEY (shown with the „D option on the KEY or KEY1 keyword) and the compare routines find a set of KEYs on a record from file SYSUT2 that is higher than the next set of KEYs on a record from file SYSUT1, COMPAREX sends the SYSUT2 record to the Differences Report, showing it as a KEY synchronization mismatch with message CPX62I. If COPYDIFF is in effect, COMPAREX sends the SYSUT2 record to the output processing routines for writing to file SYSUT3.

SEGMENT Synchronization

If the user specified one or more SEGMENT keywords and COMPAREX found no KEY keywords, COMPAREX uses SEGMENT synchronization to pair records to send to the compare routines.

If both SEGMENT and KEY keywords have been specified, COMPAREX issues message CPX18I and continues to process, using only the KEY keywords.

SEGMENTS are much like KEYs except that the input files, with SEGMENTS, are expected to be databases. The matching process is much the same, and the details are repeated here.

SEGMENTS are file control fields. The first part of the information in the SEGMENT keyword tells COMPAREX how to identify the SEGMENT by specifying a logical test. If a control field is associated with the SEGMENT, it is identified in the second part of the SEGMENT keyword.

An example of the use of SEGMENTS would be for a customer database. The first SEGMENT would define the customer header record, the second SEGMENT would be for the invoice records associated with that customer, and the third SEGMENT would be for the transactions associated with each invoice.

The SEGMENT keywords could look like this:

```
SEGMENT=(1,EQ,C'CUSTHDR',(R,9,5))
SEGMENT=(1,EQ,C'INVOICE',(D,14,4))
SEGMENT=(1,EQ,C'TRANS',(A,18,6))
```

The SEGMENT synchronization processing is as follows:

Database Out of Sequence

If, while reading records from either SYSUT1 or SYSUT2, COMPAREX finds that a KEY or SEGMENT control field is not in the sequence that has been specified (Ascending or Descending), then message CPX36A will be issued once followed by the offending record. For example, if within a group of records with a common SEGMENT identifier, the SEGMENT control fields are 1, 5, 10, 3, and 20; the record with SEGMENT control field 3 is out of synch (Ascending assumed) with the rest. It would be noted with message CPX36A and printed, but any future "out-of-synch" situations would not be flagged - only the first violation.

If it is known in advance that DATA records are not in any pure ascending or descending sequence, then Random KEYs or SEGMENT control fields should be considered.

Duplicate SEGMENT on Same File

If the compare routines find that a SEGMENT with a control field is equal to the last SEGMENTS control field from that file, COMPAREX attempts to match this SEGMENT to a SEGMENT from the other file. Equal SEGMENTS, then, cause no problems with synchronization, except that if only one SEGMENT for that control field exists on the other file, it will be paired for comparison to the first of the duplicate SEGMENTS and the second record on the file with the duplicate SEGMENT will be identified as a Segmenting Synchronization mismatch, showing on the Differences Report after message CPX64I (for records from file SYSUT1) or CPX65I (for records from file SYSUT2).

SEGMENT Starts Beyond Segment Length

If the starting position of any SEGMENT control field goes beyond the end of the record, COMPAREX will terminate immediately, issuing message CPX37A with a return code of 16.

End Of Data on SYSUT1

If file SYSUT1 has come to end of file and file SYSUT2 has not come to end of file, COMPAREX sends all extra segments from file SYSUT2 to the Differences Report, showing them as extra records with message CPX57I. If COPYDIFF is in effect, COMPAREX sends the SYSUT2 records to the output processing routines for writing to file SYSUT3.

End Of Data on SYSUT2

If file SYSUT2 has come to end of file and file SYSUT1 has not come to end of file, COMPAREX sends all extra segments from file SYSUT1 to the Differences Report, showing them as extra records with message CPX56I.

Records Matched

If the compare routines find that a SEGMENT (and its control field, if any) on a record from file SYSUT1 is equal to a SEGMENT on a record from file SYSUT2 (the pair has been sent together to the compare routines, synchronized by SEGMENT), COMPAREX compares the data values in the record.

Records Are Equal

If all bytes in the two records compare equal, COMPAREX bypasses the records. The utility does not send the records to the Differences Report, and it does not send the SYSUT2 record to the output processing routines for writing to file SYSUT3.

Using FIELDS

If FIELDS have been specified or if FIELDS have been compiled from MASK statements, COMPAREX compares just the bytes defined by FIELD keywords.

FIELDS Compare Equal

If all bytes in these FIELDS compare equal, COMPAREX bypasses the records. The utility does not send the records to the Differences Report, and it does not send the SYSUT2 record to the output processing routines for writing to file SYSUT3.

Records Compare Unequal

If all bytes in these FIELDS do not compare equal or if no FIELDS have been specified and the records do not compare equal or if no FIELDS have been specified and one record is longer than the other record, COMPAREX sends both records to the Differences Report, showing the SYSUT1 record with message CPX51I and showing the SYSUT2 record with message CPX52I. If COPYDIFF is in effect, COMPAREX sends the SYSUT2 record to the output processing routines for writing to file SYSUT3.

Match On SEGMENT, Mismatch On Control Field

If the compare routines find that a SEGMENT on a record from file SYSUT1 is equal to a SEGMENT on a record from file SYSUT2 but the control fields (as specified by the optional second part of the information in the SEGMENT keyword) are not equal, COMPAREX examines the control fields.

Ascending Control Field, SYSUT1 Low

If the control field is an ascending control field and the control field on the SYSUT1 record is lower than the control field on the SYSUT2 record, COMPAREX sends the SYSUT1 record to the Differences Report, showing it as a Segmenting Synchronization Mismatch with message CPX64I.

Descending Control Field, SYSUT1 Low

If the control field is a descending control field and the control field on the SYSUT1 record is lower than the control field on the SYSUT2 record, COMPAREX sends the SYSUT2 record to the Differences Report, showing it as a Segmenting Synchronization Mismatch with message CPX65I. If COPYDIFF is in effect, COMPAREX sends the SYSUT2 record to the output processing routines for writing to file SYSUT3.

Ascending Control Field, SYSUT2 Low

If the control field is an ascending control field and the control field on the SYSUT2 record is lower than the control field on the SYSUT1 record, COMPAREX sends the SYSUT2 record to the Differences Report, showing it as a Segmenting Synchronization Mismatch with message CPX65I. If COPYDIFF is in effect, COMPAREX sends the SYSUT2 record to the output processing routines for writing to file SYSUT3.

Descending Control Field, SYSUT2 Low

If the control field is a descending control field and the control field on the SYSUT2 record is lower than the control field on the SYSUT1 record, COMPAREX sends the SYSUT1 record to the Differences Report, showing it as a Segmenting Synchronization Mismatch with message CPX64I.

No Match On SEGMENT for SYSUT1

If the compare routines find a SEGMENT from file SYSUT1 that is not equal to any SEGMENT from file SYSUT2, COMPAREX sends the SYSUT1 record to the Differences Report, showing it as a Segmenting Synchronization mismatch with message CPX64I.

No Match On SEGMENT for SYSUT2

If the compare routines find a SEGMENT from file SYSUT2 that is not equal to any SEGMENT from file SYSUT1, COMPAREX sends the SYSUT2 record to the Differences Report, showing it as a Segmenting Synchronization mismatch with message CPX65I. If COPYDIFF is in effect, COMPAREX sends the SYSUT2 record to the output processing routines for writing to file SYSUT3.

Same-Physical-Record-Number Synchronization

If the user specified no KEY keywords and no SEGMENT keywords and the SYSUT1 file organization is not ISAM or VSAM-KSDS, COMPAREX uses same-physical-record-number synchronization to pair records to send to the compare routines.

This means that COMPAREX compares each SYSUT1 record with the same numbered record on SYSUT2. Record number 1 on SYSUT1 is compared to record number 1 on SYSUT2, and record number 1001 on SYSUT1 is compared to record number 1001 on SYSUT2. If one file is longer than the other, the extra records are sent to the compare routines alone.

If FILTERs have been specified with same-physical-record-number synchronization, COMPAREX changes its procedure for sending records to the compare routines. Under same-physical-record-number synchronization, COMPAREX sends pairs of records to the compare routines unless one file is at end of file. If a record is FILTER'ed out, COMPAREX processes that records file until the utility finds a record that is FILTER'ed in; then, it sends a pair of records to the compare routines. Messages CPX51I and CPX52I show the actual input sequence record number, not the sequence number of records sent to the compare routines.

"Synchronization After FILTERing - No KEY" on page 105 shows an example. If the two input files had ten records each and FILTER keywords FILTER'ed out records 2, 4, 6, 8, and 10 from SYSUT1, COMPAREX would send pairs of records to the compare routines.

SYSUT1		SYSUT2
1	paired with	1
3	paired with	2
5	paired with	3
7	paired with	4
9	paired with	5
		6 extra record
		7 extra record
		8 extra record
		9 extra record
		10 extra record

Figure 57. Synchronization After FILTERing - No KEY

End Of Data on SYSUT1

If file SYSUT1 has come to end of file and file SYSUT2 has not come to end of file, COMPAREX sends all extra records from file SYSUT2 to the Differences Report, showing them as extra records with message CPX57I. If COPYDIFF is in effect, COMPAREX sends the SYSUT2 record to the output processing routines for writing to file SYSUT3.

End Of Data on SYSUT2

If file SYSUT2 has come to end of file and file SYSUT1 has not come to end of file, COMPAREX sends all extra records from file SYSUT2 to the Differences Report, showing them as extra records with message CPX56I.

Records Compare Equal

If the compare routines have received a record from both input files and all bytes in the two records compare equal, COMPAREX bypasses the records. The utility does not send the records to the Differences Report, and it does not send the SYSUT2 record to the output processing routines for writing to file SYSUT3.

Using FIELDS

If FIELD have been specified or if FIELDS have been compiled from MASK statements, COMPAREX compares just the bytes defined by FIELD keywords.

FIELDS Compare Equal

If all bytes in these FIELDS compare equal, COMPAREX bypasses the records. The utility does not send the records to the Differences Report, and it does not send the SYSUT2 record to the output processing routines for writing to file SYSUT3.

Records Compare Unequal

If all bytes in these FIELDS do not compare equal or if no FIELDS have been specified and the records do not compare equal or if no FIELDS have been specified and one record is longer than the other record, COMPAREX sends both records to the Differences Report, showing the SYSUT1 record with message CPX51I and showing the SYSUT2 record with message CPX52I. If COPYDIFF is in effect, COMPAREX sends the SYSUT2 record to the output processing routines for writing to file SYSUT3.

Keywords Not Available With DATA

Certain keywords may not be used with DATA comparison logic.

- FRAME is not used with DATA comparison logic since single records rather than blocks of records are identified as differing.
- MLC has no meaning with DATA comparison logic since DATA comparison logic synchronizes on record number or KEY values. If MLC is specified, it will be ignored.
- SQUEEZE has no meaning with DATA comparison logic since no characters are deleted from records prior to the comparison. If certain fields are to be ignored in the comparison, MASK keywords can be used. If SQUEEZE is specified, it will be ignored.
- TEXT and DATA are mutually exclusive; TEXT is not used if DATA comparison logic is needed.

Advantages of DATA

DATA comparison logic has the advantages, over TEXT comparison logic, of being more efficient unless Random keys are specified, and pointing out differences at the byte or nibble level.

More Efficient

DATA comparison logic takes less CPU time than TEXT comparison logic since it does not squeeze out characters, and unless Random keys are specified, it does not search through buffer areas for matches, nor move records around in buffer areas.

Points Out Differing Bytes

DATA comparison logic shows the bytes where differences occur by underscoring those bytes on the Differences Report. The user can easily locate the differences. In addition, the user can specify that differing nibbles (half-bytes) be identified on the Differences Report.

Disadvantage of DATA

When there is no way to synchronize files, DATA comparison logic does not produce a useful Differences Report.

For example, if a JCL file, with no sequence number and many insertions, were to be compared to the previous version of that JCL file, DATA comparison logic would not be able to match the records in a way that would help the user identify the changes.

Decisions About DATA and TEXT

If the user is undecided about whether to run a COMPAREX job using DATA comparison logic or one using TEXT comparison logic, these suggestions may help.

1. Decide if the files have a KEY. If a KEY is available for synchronization, DATA comparison logic will probably produce a more useful Differences Report.
2. If the files are databases, use DATA comparison logic with SEGMENT keywords.
3. If the files are load modules, use DATA comparison logic, specifying no KEY.
4. If the size of the record is large (greater than 2000 bytes) and internal blocking is used (such as VSAM-ESDS IMS databases), use DATA comparison logic with no KEY and set MAXDIFF to a low number to prevent a runaway comparison.
5. Otherwise, use TEXT, increasing BUFF and lowering MLC until the desired results are obtained.

End-Of-Job Counts

At the end of the DATA comparison logic run, COMPAREX shows counts of its processing.

CPX74I - Bytes Underscored

Message CPX74I shows the number of bytes underscored on the Differences Report as the left-hand number.

If any SYSUT2 record was longer than the SYSUT1 record to which it was paired, message CPX74I shows the number of excess bytes underscored on the Differences Report as the right-hand number.

CPX75I - Record Counts

Message CPX75I shows the number of records read from the input files and written to any SYSUT3 file. In addition, the number of differences is shown with an explanation.

CPX76I - Unusable FIELDS, IDENTITYs, SEGMENTs, DESENs

If the length of any FIELD, IDENTITY, SEGMENT, or DESEN went beyond the length of any associated input record, message CPX76I shows a count of these occurrences.

CPX78I - Member Counts

Message CPX78I shows the number of members read from the input files. In addition, the number of differences is shown with an explanation.

The DATA Keywords

The KEY, KEY1, KEY2, and SEGMENT keywords are used to direct the matching of records.

KEY

Keyword Format

```
KEY=(ddd,111[({,} )][({,A})])  
      [({,C})][({,D})]  
      [({,Z})][({,R})]  
      [({,P})]  
      [({,B})]
```

specifies a control field used for file synchronization and to determine out-of-synch conditions. Up to forty (40) KEYs (or KEY1 and KEY2 pairs) may be specified.

The most major key on the file is specified on the first KEY (or KEY1 and KEY2 pair); KEYs are used as necessary until the most minor key on the file is specified on the last KEY (or KEY1 and KEY2 pair).

C - the default - stands for Character and length cannot exceed 256; Z for Zoned and length cannot exceed 8; P for Packed and length cannot exceed 8; B for Binary and length cannot exceed 4.

A - the default - specifies Ascending; D specifies Descending; R specifies Random.

It is not recommended that you specify Zoned, Packed, and Binary in KEYs unless there is a very logical reason to do so. The reasons for this are:

1. If you specify Zoned, Packed, or Binary for a KEY that indeed is not of that format, COMPAREX willabend when attempting to compare it to its counterpart in the other file. No precautions are made to check for format consistency in order to maintain speed.
2. When no exotic (Z, P, or B) specification is made, the default is Character. These comparisons (internally done as CLC) are faster and just as accurate in determining synchronism.

If no KEY has been specified or if all KEY statements have been incorrect and the file organization for SYSUT1 uses a key (such as ISAM or VSAM-KSDS), COMPAREX will take that key for synchronization.

If KEYs (or KEY1 and KEY2 pairs) are specified, SEGMENT keywords may not be specified.

Keyword Examples

```
KEY=(3,19)  
KEY=(21,3,P,A)  
KEY=(000024,0003,B,D)
```

KEY1

Keyword Format

```
KEY1=(ddd,111[({,} )][({,A})])  
      [({,C})][({,D})]  
      [({,Z})][({,R})]  
      [({,P})]  
      [({,B})]
```

specifies a control field used for file synchronization and to determine out-of-synch conditions. The KEY1 keyword is used to show the relative position, length, and format of the field on SYSUT1, and the KEY2 keyword is used to show its counterpart on SYSUT2. The KEY2 usually follows the KEY1 keyword. When COMPAREX cannot find a paired KEY2, it changes the KEY1 to a KEY. In all other respects, KEY1 is like KEY.

Keyword Examples

```
KEY1=(3,19)
KEY1=(21,3,P,A)
KEY1=(000024,0003,B,D)
```

KEY2

Keyword Format

```
KEY2=[(]ddd[,111[{,Z}]][]]
      [{,P}]
      [{,B}]
```

specifies a control field used for file synchronization and to determine out-of-synch conditions. The key may differ in displacement, length, and format versus its associated KEY1. See KEY1 for further information about the KEY2 keyword.

Keyword Examples

```
KEY1=(3,19),KEY2=5
KEY1=(21,3,P,A),KEY2=(45,3,P)
KEY1=(000024,0003,B,D),KEY2=(2,4,B)
```

SEGMENT

Keyword Format

```
SEGMENT=(ddd,EQ,t'vvvv'[,({A},ddd,lll)])  
(or SEG)           {D}  
                   {R}
```

specifies a control field when the input files are databases. The KEY keywords (and KEY1 and KEY2 pairs) are generally not used if COMPAREX is to process databases. If any SEGMENT keyword is specified, KEY keywords may not be specified.

The order of the SEGMENT keywords tells COMPAREX about the hierarchical structure of the database. The first SEGMENT keyword describes the highest ranking segment type. For example, on a payroll database, the highest ranking segment type might be for employee identification data, and the control field on this segment might be employee number.

The next SEGMENT keyword describes the second highest ranking segment type. For example, on a payroll database, the second highest ranking segment type might be for departmental information, with each employee having one or more departments that he or she reports time to.

Then, the last SEGMENT keyword describes the lowest ranking segment type. For example, on a payroll database, the lowest ranking segment type might be for the weekly time card information.

The three SEGMENT keywords for this payroll example could be entered in this order:

```
SEGMENT=(1,EQ,C'EMPL')  
SEGMENT=(1,EQ,C'DEPT')  
SEG=(1,EQ,C'TIME',(D,50,3))
```

The first part of the variable information in the keyword (ddd,EQ,t'vvvv') tells COMPAREX how to identify a segment type. The second part of the variable information in the keyword (A or D or R, ddd,lll) is optional; it specifies the control field that is associated with the segment type. In this second part, A - the default - specifies Ascending; D specifies Descending; R specifies Random.

COMPAREX provides a table area of 1024 bytes for SEGMENT keywords. Each SEGMENT keyword entered by the user takes up thirteen (13) bytes plus the length of the value in the first part of the variable information (double if a Wildcard is used). If a control field is also specified, add four (4) more bytes. If the type of the value is character (C), each byte of the value between the apostrophes takes up one byte of the table; if the type of the value is hexadecimal (X), each byte of the value between the apostrophes takes up one-half byte of the table (an odd number of bytes is rounded up). For example:

```
SEGMENT=(1,EQ,C'SEG01',(A,71,3))  
SEG=(1,EQ,C'CHILDREN')
```

would take up 43 of the table 1024 bytes (13 for each of the two SEGMENT keywords, 4 for the control field, 5 and 8 respectively for the SEGMENT identifiers). "Hierarchical Database Structure" on page 111 shows the hierarchical structure of two databases with three segment types. A reorganizational unload of each of them would result in a similar flat file; reading top to bottom and left to right. However, the concatenated key for the STEMS segment would be different.

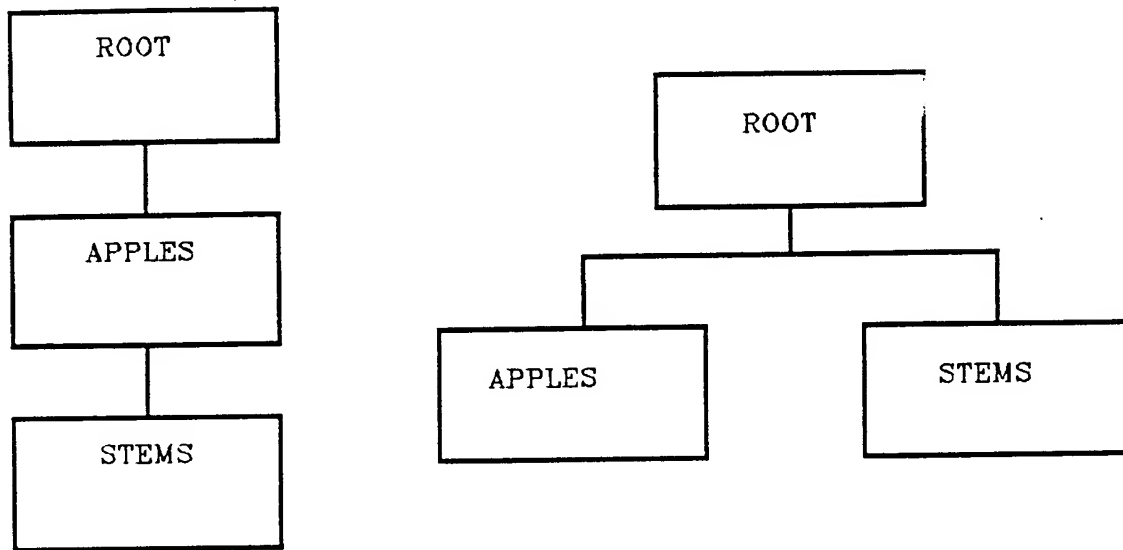


Figure 58. Hierarchical Database Structure

The database is in that hierarchical structure and ROOT has an ascending control field, APPLES has no control field, and STEMS has a random control field, then these statements:

```
SEG=(1,EQ,C'ROOT',(A,09,5))
SEGMENT=(1,EQ,C'APPLES')
SEG=(1,EQ,C'STEMS',(R,68,4))
```

could synchronize the two versions of these databases.

Random Segments are not recommended.

See "Chapter 4 - Interfaces" on page 28 for more information on using COMPAREX with databases.

Chapter 7 - TEXT Keywords

COMPAREX recognizes two categories of files - DATA and TEXT - and the utility has separate comparison logic routines for each. This chapter discusses TEXT files and the TEXT comparison logic routines; "Chapter 6 - Data File Synchronization Keywords" on page 99 discusses DATA files and the DATA comparison logic routines.

When COMPAREX finds the TEXT keyword and the utility has not nullified TEXT because of other keywords, COMPAREX uses its TEXT comparison logic routines to compare the two input files.

When COMPAREX performs TEXT comparison logic, it cannot also perform DATA comparison logic in the same run. For this reason, the TEXT keyword and the DATA keyword are mutually exclusive; they may not both be entered by the user. If both keywords are entered, COMPAREX will use the last one specified.

Text Processing Keywords

Table 3. Text Keyword Properties

Keyword	Function	Applicability		
		Data	Text	Directory
HELP = TEXT	Receive a 'canned' paragraph of cryptic keyword syntax and explanation on what it does.	-	✓	-
BUFF	Buffer size (in Kilobytes) to store records such that look-ahead logic can match records and isolate inserted/deleted records. Range 32 to 1024. Limit 16 Meg under MODE = SYS.	✓	✓	-
FRAME	Blocks of differing records are surrounded by the PLUS character and separated by the DASH character. Options are YES, NO, and NUM.	-	✓	-
MLC	Matching line count. When this many consecutive lines match, a back-in-synchronization situation allows isolation of unmatched records. Range 1 to 40.	-	✓	-
PRINT	Option FULL displays the changes in context with the entire listing of the SYSUT1 file.	✓	✓	-
SQUEEZE	Characters that are to be deleted from each record before comparison. Maximum of 40.	-	✓	-
TEXT	Invoked Text comparison logic as opposed to Data logic. Usually done for files that have no inter-record relationship such as program source code, JCL, or documentation.	-	✓	-

What is TEXT?

TEXT is defined, in COMPAREX, as any file where there is not a known inter-record relationship. TEXT files may not have bytes or fields in any fixed relationship between records. The records may contain blanks and they may be entirely free-form.

Examples of TEXT files are:

- application program source code
- job control language (JCL)
- reports
- documentation (such as this reference manual)

What is TEXT Comparison Logic?

Under TEXT comparison logic, COMPAREX does not send pairs of records to the compare routines based on KEY, SEGMENT, or same-physical-record number synchronization.

Instead, COMPAREX attempts to match records by looking ahead on each file when the record on SYSUT1 does not match the record on SYSUT2.

The order of TEXT processing is as follows:

1. First, COMPAREX deletes all SQUEEZE characters from each record as it is read. If the user has specified some option to TEXT (such as *TEXT=COBOL* or *TEXT=JCL*), COMPAREX deletes the characters specified with that option (such as *spaces and commas* with *TEXT=COBOL* or *spaces* with *TEXT=JCL*) and it will move the remaining characters to the high-order positions of the record (it will move the undeleted characters to the left). If the user has not specified any option to TEXT, the user may specify up to 40 different characters to be deleted from each record; the SQUEEZE keyword is used to specify these characters.
2. Next, COMPAREX checks to see if the current SYSUT1 record exactly matches the current SYSUT2 record. If the two records are equal, COMPAREX reads a new record on each file's buffer. If the two records are not equal, COMPAREX reads to the end of the buffer for file SYSUT2 to find a match. If a match is found (see below to read how COMPAREX uses the MLC value to determine if a match has been made), COMPAREX identifies all intervening SYSUT2 records as differing records and the utility writes these records onto the Differences Report. If a match is not found, COMPAREX identifies the SYSUT1 record as a differing record and the utility writes this record onto the Differences Report. When records are written onto the Differences Report, COMPAREX moves the remaining buffer records to the beginning of the buffer area and reads new records from the input files into the end of the buffer area.
3. If the user has specified *PRINT=FULL*, COMPAREX writes all records from file SYSUT1 onto the Differences Report, identifying those that are not matched on file SYSUT2.
4. The user is able to determine how COMPAREX will know when a match is made by the use of the MLC (*matching line count*) keyword. The default value is five; this number means that COMPAREX will not determine that a match has been made until five records in a row exactly match. If *TEXT=JCL* is specified and no MLC keyword is specified, COMPAREX sets the MLC value to 2. If *TEXT=REPORT* is specified and no MLC keyword is specified, COMPAREX sets the MLC value to 3. If the user finds that this number did not create an acceptable report, the user can override the value for subsequent executions.
5. The user is able to determine the size of the buffering areas COMPAREX works with by the use of the *BUFF* keyword. COMPAREX uses a buffer of 60 kilobytes of storage (60K) as the default. The user can change this, entering any value between 32 (for 32K) and 1024 (for 1024K). Actually, the user can specify up to 16 Meg by first setting *MODE=SYSTEMS*. In general, the larger the size of the buffer, the more CPU time COMPAREX will use for the run. A small buffer could be used for small TEXT records (such as 8 to 40 bytes) and a large buffer could be used for large TEXT records (such as 500 bytes to 32000 bytes).
6. COMPAREX manipulates its two buffers and compares records until an end-of-file indicator is detected on either file. It identifies extra records from the file that is not at end-of-file, and the utility moves these extra records to the Differences Report as differing records.
7. COMPAREX writes the Differences Report, under TEXT comparison logic, to identify differing records rather than differing fields. For this reason, the entire record is displayed without underscoring of differing bytes or half-bytes or excess bytes. By the use of the *PRINT=FULL* keyword and the *FRAME* keyword, the user can show TEXT records in several ways:

- a) The default is to show only the differing records from the two files. Those records from file SYSUT1 that are not matched to any record from file SYSUT2 are identified, on the right-hand side of the report, with the designation **DIF O N E**; and those records from file SYSUT2 that are not matched to any record from file SYSUT1 are identified with the designation **DIF T W O**. To the far right, the logical record number is shown.

When a pair of records that are exactly equal (*after any SQUEEZE characters are removed*) occurs in the blocks, those records are not designated with the **DIF** literal on the right-hand side of the report. This **DIF** literal on the right-hand side of the report designates only records that are differing.

- b) When **PRINT=FULL** is specified, the default report described above is modified so that all records from file SYSUT1 are shown in context. When records are **FRAME**'ed (*because an option to TEXT was specified or because the user entered the FRAME keyword*), the SYSUT1 records that have been matched to a SYSUT2 record are not listed inside a **FRAME** (*they are not surrounded by the DASH character and the PLUS character on the Differences Report*). Those SYSUT1 records that have been matched to a SYSUT2 record do not show the designation 'DIF' on the right-hand side of the report. "TEXT=COBOL and PRINT=FULL" on page 116 shows an example of the use of **PRINT=FULL** and **FRAME**. Note that every record from file SYSUT1 is printed.
 - c) When **FRAME** is specified or when any option to **TEXT** except for **TEXT=REPORT** is specified, blocks of differing records are surrounded by the **DASH** character and the **PLUS** character on the Differences Report. Note the "<" and ">" characters on the **FRAME** that signify the bounds of the **FIELD** that was used in the **TEXT** comparison. "TEXT=COBOL" on page 115 shows another example of **TEXT** but without **PRINT=FULL**.
8. In general, when the differences between SYSUT1 and SYSUT2 exceed forty (40%) per cent, the usefulness of the Differences Report is questionable. At that rate of difference, the changes are considered major. If you still insist on comparing the two files, increase the **BUFF** size (**BUFF=500**) and start lowering **MLC** (**MLC=4**) to get satisfactory results.

Text Examples

The two figures that follow are comparisons of two versions of the same COBOL program as given in "COBOL1 Before Change" on page 187 and "COBOL1 After Change" on page 188.

```
+++++|+++<+1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7>+..++++8+++++
+ 002100 02 ONLY-REST-OF-REC PIC X(100). 00002100 DIF O N E 21 +
-----|-----1-----2-----3-----4-----5-----6-----7-----8-----
+ 002100 02 ONLY-REST-OF-REC. 00002100 DIF T W O 21 +
+ 002200 05 ONLY-DISP PIC XXX. 00002200 DIF T W O 22 +
+ 002300 05 ONLY-UNIT PIC X(8). 00002300 DIF T W O 23 +
+ 002400 05 ONLY-VOL PIC X(6). 00002400 DIF T W O 24 +
+ 002500 05 FILLER PIC X(83). 00002500 DIF T W O 25 +
+++++|+++<+1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7>+..++++8+++++
+ 003800 ELSE IF UPDATE-REQUEST PERFORM DO-THE-UPDATE 00003800 DIF O N E 38 +
+++++|+++<+1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7>+..++++8+++++
+++++|+++<+1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7>+..++++8+++++
+ 004600 MOVE 0 TO RETURN-CODE 00004600 DIF O N E 46 +
-----|-----1-----2-----3-----4-----5-----6-----7-----8-----
+ 004900 MOVE ZERO TO RETURN-CODE 00004900 DIF T W O 49 +
+++++|+++<+1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7>+..++++8+++++

CPX71I - END OF TEXT ON FILE SYSUT1
CPX72I - END OF TEXT ON FILE SYSUT2
CPX75I - RECORDS PROCESSED: SYSUT1(60)/SYSUT2(63),DIFFERENCES(2,1,4)
        EXPLANATION - 2 RECORDS DIFFER THAT SYNCHRONIZED TOGETHER
                      1 RECORD WAS CONSIDERED INSERTED ON SYSUT1
                      4 RECORDS WERE CONSIDERED INSERTED ON SYSUT2

CPX80I - TIME OF DAY AT END OF JOB: 10:27:37 - CONDITION CODE ON EXIT: 4
```

Figure 59. TEXT=COBOL

```
000100 IDENTIFICATION DIVISION. 00000100 O N E 1
000200 PROGRAM-ID. COBOL01. 00000200 O N E 2
000300 ENVIRONMENT DIVISION. 00000300 O N E 3
000400 INPUT-OUTPUT SECTION. 00000400 O N E 4
000500 FILE-CONTROL. 00000500 O N E 5
000600 SELECT ONLY-FILE, 00000600 O N E 6
000700 ASSIGN VSAMFILE, 00000700 O N E 7
000800 ORGANIZATION IS INDEXED, 00000800 O N E 8
000900 ACCESS DYNAMIC, 00000900 O N E 9
001000 RECORD KEY IS ONLY-KEY, 00001000 O N E 10
001100 FILE STATUS IS ONLY-FILE-STAT. 00001100 O N E 11
001200 DATA DIVISION. 00001200 O N E 12
001300 FILE SECTION. 00001300 O N E 13
001400 FD ONLY-FILE. 00001400 O N E 14
001500 01 ONLY-REC. 00001500 O N E 15
001600 02 ONLY-KEY. 00001600 O N E 16
001700 03 ONLY-ACCOUNT PIC X(10). 00001700 O N E 17
001800 03 ONLY-TYPE PIC XX. 00001800 O N E 18
001900 03 ONLY-DSN PIC X(44) OCCURS 2. 00001900 O N E 19
002000 03 ONLY-MEMBER PIC X(10) OCCURS 2. 00002000 O N E 20

+++++|+++<+1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7>+..++++8+++++
+ 002100 02 ONLY-REST-OF-REC PIC X(100). 00002100 DIF O N E 21 +
```

```

-----|-----1-----2-----3-----4-----5-----6-----7-----8-----
+ 002100 02 ONLY-REST-OF-REC. 00002100 DIF T W O 21 +
+ 002200 05 ONLY-DISP PIC XXX. 00002200 DIF T W O 22 +
+ 002300 05 ONLY-UNIT PIC X(8). 00002300 DIF T W O 23 +
+ 002400 05 ONLY-VOL PIC X(6). 00002400 DIF T W O 24 +
+ 002500 05 FILLER PIC X(83). 00002500 DIF T W O 25 +
+++++|+++.*<+++1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7>+..++++8+++++
002200 WORKING-STORAGE SECTION. 00002200 O N E 22
002300 77 ONLY-FILE-STAT PIC XX. 00002300 O N E 23
002400 01 SWITCHES. 00002400 O N E 24
002500 02 END-OF-ONLY-FILE-SW PIC X. 00002500 O N E 25
002600 88 END-OF-ONLY-FILE VALUE 'Y'. 00002600 O N E 26
002700 LINKAGE SECTION. 00002700 O N E 27
002800 01 LS-FUNCTION PIC X(8). 00002800 O N E 28
002900 88 OPEN-REQUEST VALUE 'OPEN'. 00002900 O N E 29
003000 88 READSEQ-REQUEST VALUE 'READSEQ'. 00003000 O N E 30
003100 88 CLOSE-REQUEST VALUE 'CLOSE'. 00003100 O N E 31
003200 01 LS-ONLY-REC PIC X(220). 00003200 O N E 32
003300 EJECT 00003300 O N E 33
003400 PROCEDURE DIVISION USING LS-FUNCTION, LS-ONLY-REC. 00003400 O N E 34
003500 MAIN-LINE. 00003500 O N E 35
003600 IF OPEN-REQUEST PERFORM DO-THE-OPEN 00003600 O N E 36
003700 ELSE IF READSEQ-REQUEST PERFORM DO-THE-SEQ-READ 00003700 O N E 37
+++++|+++.*<+++1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7>+..++++8+++++
+ 003800 ELSE IF UPDATE-REQUEST PERFORM DO-THE-UPDATE 00003800 DIF O N E 38 +
+++++|+++.*<+++1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7>+..++++8+++++
003900 ELSE IF CLOSE-REQUEST PERFORM DO-THE-CLOSE 00003900 O N E 39
004000 ELSE DISPLAY 'INVALID I/O FUNCTION REQUESTED' 00004000 O N E 40
004100 MOVE 12 TO RETURN-CODE. 00004100 O N E 41
004200 GOBACK. 00004200 O N E 42
004300 DO-THE-OPEN. 00004300 O N E 43
004400 OPEN I-O ONLY-FILE. 00004400 O N E 44
004500 IF ONLY-FILE-STAT = '00' 00004500 O N E 45
+++++|+++.*<+++1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7>+..++++8+++++
+ 004600 MOVE 0 TO RETURN-CODE 00004600 DIF O N E 46 +
-----|-----1-----2-----3-----4-----5-----6-----7-----8-----
+ 004900 MOVE ZERO TO RETURN-CODE 00004900 DIF T W O 49 +
+++++|+++.*<+++1++++.++++2++++.++++3++++.++++4++++.++++5++++.++++6++++.++++7>+..++++8+++++
004700 ELSE 00004700 O N E 47
004800 EXHIBIT NAMED ONLY-FILE-STAT 00004800 O N E 48
004900 DISPLAY 'OPEN FAILED' 00004900 O N E 49
005000 MOVE 8 TO RETURN-CODE. 00005000 O N E 50
005100 DO-THE-SEQ-READ. 00005100 O N E 51
005200 READ ONLY-FILE NEXT, AT END MOVE 8 TO RETURN-CODE. 00005200 O N E 52
005300 IF ONLY-FILE-STAT = '00' 00005300 O N E 53
005400 MOVE ONLY-REC TO LS-ONLY-REC 00005400 O N E 54
005500 MOVE 'N' TO END-OF-ONLY-FILE-SW 00005500 O N E 55
005600 ELSE 00005600 O N E 56
005700 MOVE 'Y' TO END-OF-ONLY-FILE-SW 00005700 O N E 57
005800 MOVE 8 TO RETURN-CODE. 00005800 O N E 58
005900 DO-THE-CLOSE. 00005900 O N E 59
006000 CLOSE ONLY-FILE. 00006000 O N E 60
CPX71I - END OF TEXT ON FILE SYSUT1
CPX72I - END OF TEXT ON FILE SYSUT2
CPX75I - RECORDS PROCESSED: SYSUT1(60)/SYSUT2(63),DIFFERENCES(2,1,4)
EXPLANATION - 2 RECORDS DIFFER THAT SYNCHRONIZED TOGETHER
1 RECORD WAS CONSIDERED INSERTED ON SYSUT1
4 RECORDS WERE CONSIDERED INSERTED ON SYSUT2
CPX80I - TIME OF DAY AT END OF JOB: 10:27:42 - CONDITION CODE ON EXIT: 4

```

Figure 60. TEXT = COBOL and PRINT = FULL

Keywords Not Available With TEXT

Certain keywords may not be used with TEXT comparison logic.

Dummy File

SYSUT2=DUMMY may not be specified with TEXT. COMPAREX, in order to do TEXT comparison, needs a valid file as SYSUT2. If the user specifies TEXT and SYSUT2=DUMMY, COMPAREX issues message CPX24A and continues to process, ignoring the TEXT keyword and using DATA comparison logic for the run.

KEYs

KEY keywords (*and KEY1 and KEY2 pairs*) may not be specified with TEXT. COMPAREX does not synchronize on KEYs with TEXT. If the user specifies any KEY and TEXT, COMPAREX issues message CPX24A and continues to process, ignoring the TEXT keyword and using DATA comparison logic for the run, synchronizing by KEY.

SEGMENTs

SEGMENT keywords may not be specified with TEXT. COMPAREX does not synchronize on SEGMENTs with TEXT. If the user specifies any SEGMENT and TEXT, COMPAREX issues message CPX24A and continues to process, ignoring the TEXT keyword and using DATA comparison logic for the run, synchronizing by SEGMENT.

DATA

DATA may not be specified with TEXT; the two are mutually exclusive. If both keywords are specified, COMPAREX will use the last one.

IDENTITYs

IDENTITY keywords may not be specified with TEXT. TEXT comparison logic does not use IDENTITY tests. If the user specifies any IDENTITY and TEXT, COMPAREX issues message CPX24A and continues to process, ignoring the TEXT keyword and using DATA comparison logic for the run.

FIELDs

More than one FIELD keyword (*and FIELD1 and FIELD2 pair*) may not be specified with TEXT. If any option to TEXT has been specified (*such as TEXT=COBOL or TEXT=JCL*), no FIELD may be entered since COMPAREX creates a FIELD when it processes with an option to TEXT (*when MODE=APPLICATIONS is in effect, for TEXT=COBOL, COMPAREX creates FIELD=(7,66) and FIELD=(1,72) for TEXT=JCL*). If the user enters more than one FIELD with TEXT, COMPAREX issues message CPX24A and continues to process, ignoring the TEXT keyword and using DATA comparison logic.

DESENs

The desensitizing keywords (*DESEN, DESEN1, and DESEN2*) may not be used with TEXT. They will be ignored, if entered.

MASKs

If a MASK keyword generates only one FIELD, it may be used with TEXT. If more than one FIELD is generated, COMPAREX ignores the TEXT keyword. See "Chapter 5 - Input Processing Keywords" on page 69 for information about how COMPAREX generates FIELDs from MASKs.

DIRECTORY

DIRECTORY is not used with TEXT.

FLDSONLY

FLDSONLY may not be specified with TEXT. COMPAREX does not underscore any bytes on the Differences Report with TEXT. COMPAREX will ignore this keyword if TEXT comparison logic is in effect.

LINE

If any option to TEXT processing except for TEXT=REPORT is specified (such as TEXT=COBOL or TEXT=JCL), COMPAREX sets the value of LINE to (80,ALPHA) and any LINE keyword entered by the user will be ignored. If TEXT=REPORT is specified, COMPAREX sets the value of LINE to (110,ALPHA). If no option to TEXT processing is specified, COMPAREX defaults the value of LINE to (32,HORIZONTAL) and the user may modify this by specifying any other legitimate value for LINE.

NIBBLE

NIBBLE may not be specified with TEXT. COMPAREX does not underscore half-bytes on the Differences Report with TEXT. COMPAREX will ignore this keyword if TEXT comparison logic is in effect.

PRINTs

PRINT=MATCH, PRINT=NOMATCH, PRINT=MISMATCH, and PRINT=NOMISMA CH may not be specified with TEXT. COMPAREX will ignore these keywords if TEXT comparison logic is in effect. PRINT=FULL is the only PRINT keyword that may be entered with TEXT; this specifies that COMPAREX will print all records from file SYSUT1 along with all differing records from file SYSUT2 on the Differences Report.

Advantages Of TEXT

TEXT comparison logic has the advantages, over DATA comparison logic, of comparing only on the significant data, determining differences in context with the rest of the file, and providing synchronization when no KEY is clear.

Comparing Only On The Significant Data

Before any comparison takes place under TEXT processing, COMPAREX removes insignificant characters through its SQUEEZE process. What is left is the significant data, and only this data enters the comparison process.

An example would be the common practice of columnar indentation in COBOL programs. This columnar indentation creates spaces in the input file, and COMPAREX removes these spaces before comparison. In this way, if TEXT=COBOL were specified, COMPAREX would find these two records to be equal:

```
010100      IF SORT-RETURN IS NOT EQUAL TO ZERO,  
010111      IF      SORT-RETURN  IS      NOT EQUAL TO      ZERO
```

COMPAREX would remove all spaces and commas, and it would compare only on positions 7 through 72 (relative to one).

Determining Differences In Context With The Rest Of The File

By specifying PRINT=FULL, the user can inspect all inserted, deleted, and modified records while examining the rest of the file.

In this way, some errors that could not be detected by viewing just differing records can be more clearly seen.

Providing Synchronization When No KEY Is Clear

For files without a specific synchronization KEY, TEXT comparison can be used to identify inserted, deleted, or changed records. TEXT comparison logic provides the user with a way to match up records according to data values when the files cannot be synchronized by KEY.

Disadvantages Of TEXT

TEXT comparison logic has disadvantages. It is inefficient, and it does not point out differing bytes or half-bytes.

Inefficient

TEXT comparison logic requires more CPU time than DATA comparison logic because it squeezes out specific characters, it may access its buffer areas many times during the comparison of one record, and it manipulates its buffer areas to move records into lower storage when new records are added to the buffer.

Does Not Point Out Differing Bytes

TEXT comparison logic identifies differing records. It produces a Differences Report showing the full record, but differing bytes and half-bytes are not underscored and excess bytes are not underscored.

Decisions About DATA and TEXT

If the user is undecided about whether to run a COMPAREX job using DATA comparison logic or one using TEXT comparison logic, these suggestions may help.

1. Decide if the files have a KEY. If a KEY is available for synchronization, DATA comparison logic will probably produce a more useful Differences Report.
2. If the files are databases or previously unloaded versions (flat files), use DATA comparison logic with SEGMENT keywords.
3. If the files are load modules, use DATA comparison logic, specifying no KEY.
4. If the size of the record is large (greater than 2000 bytes) and internal blocking is used (such as VSAM-ESDS IMS databases), use DATA comparison logic with no KEY and set MAXDIFF to a low number to prevent a runaway comparison.
5. Otherwise, use TEXT, increasing BUFF and lowering MLC until the desired results are obtained.

End-of-Job Counts

At the end of the TEXT processing, COMPAREX shows with message CPX75I the number of records read from the input files and written to any SYSUT3 file.

In addition, the number of differences are shown with message CPX75I.

The TEXT Keywords

The BUFF, FRAME, MLC, SQUEEZE, TEXT, and PRINT=FULL keywords are used to specify TEXT comparison logic and to direct the matching, buffering, and printing routines.

BUFF

Keyword Format

```
BUFF={ 60 }  
      { nn }
```

specifies the number of kilobytes for buffering records. nn may be a value as low as 32 and as high as 1024 under MODE=APPLICATIONS, and as high as 16384 under MODE=SYSTEMS. The default value for TEXT processing, or DATA with Random KEY's is 60. For csect parsing (DATA=CSECT), the default buffer size is 256K. It is recommended that the user specify much larger than that, particularly if it is known that the load modules to be compared are large.

For example:

```
MODE=SYS      /* Temporarily set SYStems mode */  
BUFF=5000     /* Five meg buffer - if possible */  
MODE=APL      /* Reset back to AppLications */
```

Setting the size of the buffer within these keywords does not necessarily mean that the GETMAIN to obtain the storage will be satisfied. If the region is not large enough to handle such a large request, the task may abend or receive the following message:

```
CPX99A - INSUFFICIENT VIRTUAL STORAGE - FUNCTION TERMINATED
```

and terminate immediately.

The reasons for this larger BUFFER are primarily because of the need to process databases that contain random keys. CSECT Parsing will, on occasion, also need extremely large BUFFER's. The more buffer storage supplied, the higher the probability that a record with a random key in the SYSUT1 half of the buffer will match a random key in the SYSUT2 half of the buffer. Some TEXT type files (files that are processed using TEXT logic) also need this larger buffer because of their sheer record size. 4000-byte (four K) records will fill up a 60K buffer very quickly. The specification of a very large value, such as 1024, should be done only when processing a very large TEXT record (a record exceeding 1,000 bytes) or when a large number of clustered inserts are included in either file. In general, the larger the buffer, the more processing time COMPAREX will take.

Internally, this buffer is split in half. One half of the buffer holds records from SYSUT1, and the other half of the buffer holds records from SYSUT2.

Keyword Examples

```
BUFF=512  
BUFF=1024 /* 1K
```


FRAME

Keyword Format

```
FRAME [= { NUM } ]  
      [ { YES } ]  
      [ { NO } ]
```

specifies that blocks of differing records are surrounded by the PLUS character and the DASH character on the Differences Report.

If any option to TEXT is specified, except for TEXT=REPORT, the blocks of differing records are automatically FRAME'ed with FRAME=NUM on the Differences Report. FRAME=YES and FRAME are logically equivalent and direct that the FRAME is composed solely of PLUS'es and DASH'es. FRAME=NUM intersperses numerics into the FRAME to show relative displacements. FRAME=NO overrides any other specification or defaults and effectively nullifies the FRAME construction.

Keyword Examples

```
FRAME  
FRAME=NO  
FRAME= (NUM)
```

MLC

Keyword Format

```
MLC={ 05 }  
      { nn }
```

specifies matching line count. nn may be a value between 1 and 40. This value specifies how many records COMPAREX must match consecutively after a mismatch to determine if the files are back in synchronization. Specification of a very large value, such as 40, should be done only when the user is certain that there are very few differences. In general, the larger the matching line count, the more processing time COMPAREX will take.

If TEXT=JCL is specified and no MLC is specified, COMPAREX sets the MLC value to 2. If TEXT=REPORT is specified and no MLC keyword is entered, COMPAREX sets the MLC value to 3. Otherwise, if no MLC keyword is entered, COMPAREX sets the MLC value to 5.

Keyword Examples

```
MLC=10  
MLC= (005)
```

SQUEEZE

Keyword Format

```
SQUEEZE=t'vv'  
(or SQZ)
```

specifies the characters to be deleted from each TEXT record before comparison. The specified characters are removed, and the remaining text is moved to the 'left' in the record. Up to 40 different SQUEEZE characters may be specified.

The user may not specify a SQUEEZE keyword if any option to TEXT is specified. If an option to TEXT is specified (such as TEXT=COBOL), any SQUEEZE keywords will be ignored.

Keyword Examples

```
SQUEEZE=C ' '
SQUEEZE=X 'EA'
SQZ=C'?'
```

TEXT

Keyword Format

```
TEXT=[ $ ] { ALC }      ]
      [ { BAL }         ]
      [ { C }           ]
      [ { CLIST }       ]
      [ { COBOL }       ]
      [ { FORTRAN }     ]
      [ { JCL }         ]
      [ { PANEL }       ]
      [ { PASCAL }      ]
      [ { PL1 }         ]
      [ { PL/1 }        ]
      [ { PLI }         ]
      [ { PL/I }        ]
      [ { REPORT }     ]
      [ { RPG }         ]
      [ { SCRIPT }     ]
      [ { . }          ]      <=== Whatever WILDCARD is set to
```

specifies that files have no interrecord relationship (such as program source code). Inserted, deleted, and modified records are isolated in the Differences Report, but differing bytes are not underscored.

If any option to TEXT processing is specified (except for TEXT=REPORT or TEXT=SCRIPT), LINE is set to LINE=(80,ALPHA). This gives a Differences Report in character format, not the hexadecimal dump format.

It is not necessary to specify any option (programming language). If the TEXT keyword is used without an option, COMPAREX will SQUEEZE out only those characters specified by the SQUEEZE keyword, and it will compare the entire record (unless a FIELD is specified).

None of the data files synchronization keywords (KEY, KEY1, KEY2, and SEGMENT) may be used with TEXT processing.

OPTIONS:

\$ - as a prefix to any of the TEXT options, such as TEXT=\$COBOL, nullifies any potential SQUEEZEing.

ALC or BAL or JCL or PASCAL or C - COMPAREX will SQUEEZE out spaces. Under the default mode (displacement relative to one), the utility will compare only on positions 1 through 72 of the record. If the user specifies MODE=SYSTEMS (displacement relative to zero), the utility will compare only on positions 0 through 71. If the JCL option is specified and the user did not enter an MLC keyword, MLC is set to 2.

CLIST - COMPAREX will SQUEEZE out spaces. This option assumes that the files to be compared are variable length CLISTs. Fixed length CLISTs should be

compared using TEXT=BAL. Under the default mode (displacement relative to one), the utility will compare on position 9 through the END of the record. If the user specifies MODE=SYSTEMS (displacement relative to zero), the utility will compare on positions 12 through the END of the record displaying the LLBB and 8-digit sequence number.

COBOL - COMPAREX will SQUEEZE out spaces and commas. Under the default mode (displacement relative to one), the utility will compare only on positions 7 through 72 of the record. If the user specifies MODE=SYSTEMS (displacement relative to zero), the utility will compare only on positions 6 through 71.

FORTRAN or PL/1 or PL/I or PL1 or PLI - COMPAREX will SQUEEZE out spaces. Under the default mode (displacement relative to one), the utility will compare only on positions 2 through 72 of the record. If the user specifies MODE=SYSTEMS (displacement relative to zero), the utility will compare only on positions 1 through 71.

PANEL - COMPAREX will not SQUEEZE out any characters (TEXT=\$PANEL is redundant). All bytes in all records are compared.

REPORT or SCRIPT - COMPAREX will not SQUEEZE out any characters, and it will compare the entire record. LINE is set to (110,ALPHA); COMPAREX displays bytes in excess of 110 only if one or more of these excess bytes are not spaces. If the user did not enter an MLC keyword, MLC is set to 3.

RPG - COMPAREX will SQUEEZE out spaces. Under the default mode (displacement relative to one), the utility will compare only on positions 2 through 66 of the record. If the user specifies MODE=SYSTEMS (displacement relative to zero), the utility will compare only on positions 1 through 65.

Wildcard (.) - COMPAREX will read the first four records from file SYSUT1 and determine what kind of TEXT compare should be run. This is certainly not foolproof, but most kinds of program source code can be determined and the appropriate FIELD deduced.

Keyword Examples

TEXT
TEXT=\$COBOL
TEXT=.

PRINT

Keyword Format

PRINT=FULL

specifies printing all records from file SYSUT1 in context with the differing records from file SYSUT2. This is recommended for comparing TEXT files that you are not very familiar with.

Chapter 8 - Output Processing Keywords

The output processing keywords direct COMPAREX to write file SYSUT3.

Output Processing Keywords

Table 4. Output Keyword Properties

Keyword	Function	Applicability		
		Data	Text	Directory
HELP=OUTPUT	Receive a 'canned' paragraph of cryptic keyword syntax and explanation on what it does.	✓	✓	✓
COPYDIFF	Differing records from SYSUT2 (as compared to SYSUT1) are written to SYSUT3. Specialized formatting for change control records are possible for: PANVALET, LIBRARIAN, GEM, IE-BUPDTE, and MEMBER.	✓	✓	✓
INSERT	Change control directive for 'inserted' records when COPYDIFF=OTH.	-	✓	-
DELETE	Change control directive for 'deleted' records when COPYDIFF=OTH.	-	✓	-
REPLACE	Change control directive for 'replaced' records when COPYDIFF=OTH.	-	✓	-
SYSUT3	Description of what file SYSUT3 looks like.	✓	✓	✓

Writing a File of Selected Records

When COPYDIFF has been specified file and the utility is able to open file SYSUT3, all differing records from input file SYSUT2 are written onto file SYSUT3.

After file SYSUT3 is successfully opened, COMPAREX issues message CPX16I to give the SYSUT3 dataset name and dataset attributes.

If COMPAREX finds the COPYDIFF keyword but the utility is unable to open file SYSUT3, COMPAREX issues message CPX16A and continues to process as if COPYDIFF had not been specified.

If MAXDIFF has been specified without CONTINUE, COMPAREX stops writing records onto SYSUT3 as soon as the MAXDIFF number is reached.

If MAXDIFF has been specified with CONTINUE, COMPAREX continues to compare records after it prints the maximum number of records on the Differences Report, and it also continues to write differing records from file SYSUT2 onto file SYSUT3.

The SYSUT3 file can be created with either DATA file comparison logic or TEXT file comparison logic.

The value of the PRINT keyword does not affect the writing of file SYSUT3. For example, if PRINT=NOMISMATCH were specified, COMPAREX would not print inserted records from the input files onto the Differences Report. Any in-

sorted records from file SYSUT2 would still be written to file SYSUT3 if COPYDIFF were specified since they are differing records.

Differing Records Defined

The records from file SYSUT2 that are differing are written onto file SYSUT3. Differing records are identified as those records that meet one or more of these tests:

SYSUT1 Is A Dummy File

If the user specified SYSUT1=DUMMY, or the JCL specified a null file for file SYSUT1 (//SYSUT1 DD DUMMY), all SYSUT2 records that pass FILTER-type tests are identified as differing.

HINT: To create a test file of selected records where account number contains the set of digits '06' in the first byte and '5' in the third byte, use these keywords:

```
SYSUT1=DUMMY /* SYSUT1 is DUMMY, SYSUT2 points to file */
COPYDIFF      /* Write selected records to SYSUT3 */
MAXDIFF=0,CONTINUE /* No need to print the data */
WILDCARD=C '*' /* Reset Wildcard character */
FILTERIN=(9,EQ,X'06**5*') /* Select specific set */
```

KEY Mismatch

If at least one KEY has been specified or if SYSUT1 is either an ISAM or VSAM/KSDS file, KEY synchronization is used. COMPAREX matches records by KEY, and any record on file SYSUT2 that is unmatched by KEY to a record on file SYSUT1 is identified as differing.

SEGMENT Mismatch

If at least one SEGMENT has been specified, SEGMENT synchronization is used. COMPAREX matches records by SEGMENT, and any record on file SYSUT2 that is unmatched by SEGMENT to a record on file SYSUT1 is identified as differing.

Extra Records On End Of SYSUT2

If COMPAREX has found neither KEY nor SEGMENT keywords, COMPAREX matches records by same physical-record number synchronization. If file SYSUT2 is longer than file SYSUT1, the extra records on file SYSUT2 are identified as differing.

SYSUT2 Record Is Longer Than SYSUT1 Record

If the SYSUT2 record is longer than its paired SYSUT1 record and one or more of the excess bytes on the SYSUT2 record are some value other than hexadecimal zero (X'00'), the SYSUT2 record is identified as differing.

FIELDS Are Differing

If FIELD comparison is being done and any byte within the FIELDS is different on the two records, the SYSUT2 record is identified as differing.

With TEXT Comparison

If TEXT file comparison logic is being used and any record on file SYSUT2 is not matched exactly to a record on file SYSUT1, that SYSUT2 record is identified as differing.

No FIELDS

If DATA file comparison logic is being used and FIELD comparison is not being done and any byte is different on the synchronized records, the SYSUT2 record is identified as differing.

With FILTERs

If FILTERs are being used and these FILTERs cause some or all of the records from file SYSUT1 to be not sent to the comparison routines, the records from file SYSUT2 to which these SYSUT1 records would otherwise be paired for comparison are identified as differing.

End-of-Job Record Count

COMPAREX shows the number of records written onto SYSUT3 with message CPX75I at the end of processing.

If COMPAREX was unable to open file SYSUT3, this number does not appear with message CPX75I.

If COMPAREX was able to open file SYSUT3 but no records were selected to be written to file SYSUT3, COMPAREX will close the file and will show the record count of zero for file SYSUT3 with message CPX75I.

The COPYDIFF Keywords

COPYDIFF

Keyword Format

```
COPYDIFF=[(<PAN> ) [,STAMP=<NO> ] [,VERS=<YES> ) [,PASS=<YES>] [,TEMP=<NO>] [,RESEQ=<NO> )]
[ (<LIB> ) [ (<YES>)[ (<NO> )] [ (<NO> ) [ (<YES>)[ (<YES>]
[ (<OTH> ) [ (<YESHMM>)]
[ (<MEMBER> )
[ (<GEM> )
[ (<IEBUPDTE>)] [,SEQFLD='ddl [,ddl]' ( )]
```

specifies that differing records from file SYSUT2 as compared to records from file SYSUT1 are written to file SYSUT3. Differing records are defined as inserted records (as determined by KEY or SEGMENT synchronization), unequal comparisons (as determined by full record compares, by the use of FIELDS, or by TEXT comparison logic), or extra records on the end of file SYSUT2.

In the special case of COPYDIFF=MEMBER in conjunction with DIR or DIRECTORY, just the member names of the two directory-embedded datasets are written to file SYSUT3.

If COPYDIFF is specified and SYSUT1=DUMMY is specified, or SYSUT1 points to a null file, all records on SYSUT2 that pass FILTER-type tests will be written onto SYSUT3.

SYSUT3 must be QSAM, VSAM, ISAM, or DUMMY. If COPYDIFF is specified and COMPAREX cannot open SYSUT3, COMPAREX issues action message CPX16A and continues to process as if COPYDIFF had not been specified.

The SYSUT3 dataset attributes are specified through JCL.

The options to COPYDIFF (*except MEMBER*) only have meaning under TEXT processing and will be nullified if not under TEXT processing. Before being written out to

SYSUT3, differing TEXT records will be preceded by a formatted change control record. Subsequent input of this SYSUT3 file (delta deck) into the proprietary library management software will create an audit trail of the changes.

The MEMBER option to COPYDIFF only has meaning under DIRECTORY or DIR processing and will be nullified otherwise. Extreme care should be taken in using the IEBUPDTE option. If the SYSUT1 member does not have sequence numbers in a format recognizable by IEBUPDTE, an abend within COMPAREX is likely. Furthermore, it is not possible to create an accurate delta deck with the IEBUPDTE option if the second dataset has records inserted at the beginning (when compared to the first dataset).

SEQFLD=738 is the default. The rules for this are similar to those for IEBUPDTE. Here are the rules of specifying SEQFLD in the form SEQFLD='ddl,ddl':

dd must be a two-digit numeric in the range 01 to 78.

l must be a one-digit numeric in the range 3 to 8.

the sum of dd and l cannot exceed 81.

if two are specified, the dd of the second must be at least as large as the first.

If the specifications for SEQFLD are made to point to card columns that indeed do not contain numeric sequence numbers, COMPAREX will abend with a S0C7, and precipitate message CPX97A when SPIE catches the abnormal end. It is your responsibility to ensure that the specifications for SEQFLD are accurate.

STAMP=NO is the default. STAMP=YES means that a date/time stamp will be placed in columns 73 through 80 of each output record in the format yymmddhh. It may not be used with the IEBUPDTE option.

VERS=YES is the default. It only has applicability under LIBrarian. VERS=YES means that the month and day (mmdd) will be generated in the format:

```
-SEL member,VERS=mmdd
```

VERS=YESHHMM means that the month, day, hour, and minute will be generated in the format:

```
-SEL member,VERS=mmddhhmm
```

PASS=YES is the default. It only has applicability under LIBrarian. PASS=YES means that the password for the member will be generated in the format:

```
-SEL member,pass
```

TEMP=NO is the default. It only has applicability under PANvalet, LIBrarian, and GEM. TEMP=YES means that the TEMP option on the member selection format card image is taken in the format:

```
-SEL member,pass,VERS=mmdd,TEMP
or
++UPDATE member,3,TEMP
or
- UPDATE member,TEMP
```

RESEQ=NO is the default. It is only valid for LIBrarian or GEM. RESEQ=YES means that the member will be renumbered when the "Delta Deck" is used.

Here are some examples of proper syntax in specifying the RESEQ option to COPYDIFF:

```
COPYDIFF=(LIB,TEMP=YES,VERS=NO,STAMP=YES)
COPYDIFF=(LIB,RESEQ=NO)
COPYDIFF=(GEM,RESEQ=YES)
```

The SYSUT3 file generated might have the header record for each updated member look like this:

```
-SEL      libmembr,pass,RESEQ  
- UPDATE  gemmember,NUM
```

Keyword Examples

```
COPYDIFF  
COPYDIFF=PAN  
COPYDIFF=(LIB,VERS=NO,PASS=NO)  
COPYDIFF=(GEM,STAMP=YES,RESEQ=YES)  
COPYDIFF=(IEBUPDTE,SEQFLD='738,783')
```

INSERT

Keyword Format

```
INSERT={ C'xxx' }  
      { xxx }
```

may be entered only if COPYDIFF=OTH. If COPYDIFF=PAN is specified, INSERT=++C is set. If COPYDIFF=LIB is specified, INSERT=-INS is set. If COPYDIFF=GEM is specified, INSERT=-INS is set. If COPYDIFF=IEBUPDTE is specified, INSERT=NUMBER is set. If COPYDIFF=MEMBER is specified, INSERT=INSERT is set. If no value for INSERT is specified, the default value is INSERT=?.

Keyword Examples

```
COPYDIFF=OTH,INSERT=C'/ INS '  
COPYDIFF=OTH,INSERT=PUT-HERE
```

DELETE

Keyword Format

```
DELETE={ C'xxx' }  
      { xxx }
```

may be entered only if COPYDIFF=OTH. If COPYDIFF=PAN is specified, DELETE=++C is set. If COPYDIFF=LIB is specified, DELETE=-DEL is set. If COPYDIFF=GEM is specified, DELETE=-DEL is set. If COPYDIFF=IEBUPDTE is specified, DELETE=DELETE is set. If COPYDIFF=MEMBER is specified, DELETE=DELETE is set. If no value for DELETE is specified, the default value is DELETE=?.

Keyword Examples

```
COPYDIFF=OTH,DELETE=C'/ DEL '  
COPYDIFF=OTH,DELETE=DEL-HERE
```


REPLACE

Keyword Format

REPLACE={ C'xxx' }
 { xxx }

may be entered only if COPYDIFF=OTH. If COPYDIFF=PAN is specified, REPLACE=++C is set. If COPYDIFF=LIB is specified, REPLACE=-REP is set. If COPYDIFF=GEM is specified, REPLACE=- REP is set. If COPYDIFF=IEBUPDTE is specified, REPLACE=CHANGE is set. If COPYDIFF=MEMBER is specified, REPLACE=REPLACE is set. If no value for REPLACE is specified, the default value is REPLACE=?.

Keyword Examples

COPYDIFF=OTH, REPLACE=C' / REP '
COPYDIFF=OTH, REPLACE=REP-HERE

SYSUT3

Keyword Format

SYSUT3={ QSAM }
 { ISAM }
 { [(] VSAM[, PASSWORD=password)] }
 { DUMMY }

overrides the dataset organization (DSORG) determined by COMPAREX for SYSUT3. This is allowed but not recommended.

Chapter 9 - Display Processing Keywords

COMPAREX generates the Differences Report to SYSPRINT. This Differences Report is a listing of the results of the comparison.

The first section of the Differences Report shows the COMPAREX license information and a listing of the comments and keywords entered by the user. If the user entered the HELP keyword, the HELP tutorial follows the listing of the user's keywords. Next, COMPAREX prints messages showing the processing parameters it uses for the execution, showing these with message numbers CPX04I through CPX19I. After these job initialization messages, COMPAREX prints any records that have been selected for printing, based on the results of the comparison routines and on the value of the PRINT keyword. Finally, COMPAREX prints its end-of- processing totals, showing these with message numbers CPX71I through CPX80I.

Display Processing Keywords

Table 5. Display Keyword Properties

Keyword	Function	Applicability		
		Data	Text	Directory
HELP=DISPLAY	Receive a 'canned' paragraph of cryptic keyword syntax and explanation on what it does.	✓	✓	✓
ASCII	Translates ASCII input into readable format. Opposite of EBCDIC.	✓	✓	-
CASE	Lower case translation - Upper or Mixed.	✓	✓	✓
DASH	1) Character used as the underscore for differing bytes of DATA. 2) Framing character separating blocks of SYSUT1 TEXT from SYSUT2 TEXT.	✓	✓	✓
DECIMAL	Relative displacement to be displayed in decimal format.	✓	✓	-
EBCDIC	Input is in EBCDIC format. Opposite of ASCII.	✓	✓	✓
FLDSONLY	Only differing bytes defined by FIELD statements are underscored with the DASH definition.	✓	-	-
FORMAT	Data formatting characteristics. Horizontal Hex, Alpha only, Vertical Hex.	✓	limited	-
GENFLDS	Generates a visual representation of all IDENTITY, FIELD, and MASK statements.	✓	✓	-
HALT	Stop execution before reading any files.	✓	✓	✓
HEX	Relative displacement to be displayed in hexadecimal format. Opposite of DECIMAL.	✓	✓	-
IGNORSIN	Ignore sign differences in packed fields.	✓	-	-

continued ...

Keyword	Function	Applicability		
		Data	Text	Directory
INTERLEAVE	Number of lines that are blocked together from a SYSUT1 record before displaying a similar number from SYSUT2.	✓	-	-
KEYSONLY	On synchronization mismatches, prints only lines completely encompassing control fields - KEYs or SEGMENTs.	✓	-	-
KILLRC	Force return code to be zero.	✓	✓	✓
KILLSPIE	Don't trap abend situations - let COMPAREX produce a dump. Generate S0C3 abend when CPXIFACE generates return code 8.	✓	✓	✓
LINE	Number of bytes displayed on each line and the method of that display.	✓	✓	-
MAXDIFF	Maximum number of differences to display.	✓	✓	✓
MBRHDR	Prints headings of members of directory-embedded datasets. Options are YES, NO, COND, and MATCH.	✓	✓	-
NIBBLE	Underscoring of half-byte nibbles by the DASH character. Only applicable to Dump format.	✓	-	-
PAGE	Number of printed lines per page.	✓	✓	✓
PLUS	1) Character used as the underscore for excess bytes when SYSUT2 DATA record is longer than SYSUT1 record. 2) Framing character surrounding blocks of TEXT.	✓	✓	-
PRINT	Directs the printing of synchronized MATCH and MISMATCH'ed records.	✓	✓	✓

The All-Defaults Differences Report

If the user enters no display processing keywords, COMPAREX produces the Differences Report, taking all defaults. Here is a listing of the values associated with the display processing keywords under the all-defaults mode:

ASCII is not in effect; EBCDIC is in effect

CASE=UPPER is in effect

DASH=C'-' is in effect

DECIMAL is in effect

EBCDIC is in effect

FLDSONLY is not in effect

FORMAT=02 is in effect

No GENFLDS are produced

No HELP listing is produced

HEX is not in effect; DECIMAL is in effect

INTERLEAVE=0 is in effect

LINE=(32,HORIZONTAL) is in effect

MAXDIFF=999999999999 is in effect

NIBBLE is not in effect

PAGE = 58 is in effect

PLUS = 'C' + ' ' is in effect

PRINT = MATCH and PRINT = MISMATCH are in effect

Note: The all-defaults job is not recommended. Every COMPAREX job should specify a MAXDIFF keyword to avoid large printouts when the expected results do not occur.

The All-Defaults Differences Report With DATA

1. COMPAREX prints its license information, showing the name and address of the computer center where the utility is installed.
2. COMPAREX acknowledges each line of keywords specified by the user. If the utility finds at least one non-blank character that is not identified as keywords, COMPAREX will underscore the characters with the DASH character and will print the literal "ERROR?" in the right-hand column. Any characters in the line beginning with message number CPX00I that are not underscored are accepted as valid keywords or comments.
3. COMPAREX prints messages showing the processing parameters it uses with the execution.
4. PAGE = 58 is used. Each page contains a maximum of 58 lines. The first two lines show the time, date, page number, and input file dataset names.
 - a) The literal ONE is shown in the right-hand column when the line contains data from a record from file SYSUT1.
 - b) The literal TWO is shown in the right-hand column when the line contains data from a record from file SYSUT2.
 - c) The literal DIFFERENCE is shown within lines of data from file SYSUT2 when the SYSUT2 record has been selected for printing on the Differences Report because of inequalities when matched to a record from file SYSUT1. This SYSUT2 record is shown with message CPX52I.
 - d) When reviewing the Differences Report, the experienced user runs his or her finger down the right-hand column, looking for the literal DIFFERENCE. These SYSUT2 records are then reviewed for reconciliation by noting the DASH character underscores (for differing bytes) and the PLUS character underscores (for extra bytes).
5. The EBCDIC translation table is used to translate the bit representations shown on the left side of the report into the printable characters on the right side of the report.
6. The displacement on each line is shown in DECIMAL. The first line shows bytes 1 through 32, the second line shown bytes 33 through 64, and the third line shows bytes 65 through 96.
7. FORMAT = 02 is used. This implies that LINE is set at (32,HORIZONTAL) for the standard IBM "dump" format. Each line contains 32 bytes of the record. In the display of the SYSUT1 record under message CPX51I, if any line is the same as the previous line, the data values are replaced by the words 'SAME AS ABOVE'. In the display of the SYSUT2 record under message CPX52I, only lines that actually have underscored differences are displayed.
8. DASH = 'C' - ' ' is used. The DASH character is shown to the left of the word 'DIFFERENCE,' and it is used to underscore differing bytes. All differing bytes are underscored on the record from file SYSUT2.
9. PLUS = 'C' + ' ' is used. The PLUS character is shown to the right of the word 'DIFFERENCE,' and it is used to underscore excess bytes when the SYSUT2 records is longer than the paired SYSUT1 record.
10. COMPAREX prints its end-of-processing totals.

Using Keywords To Modify The Differences Report

The all-defaults Differences Report can be modified by the use of COMPAREX keywords.

ASCII

translates ASCII input into readable format on the alphanumeric section of the Differences Report. ASCII stands for American Standard Code for Information Interchange. EBCDIC and ASCII are mutually exclusive. EBCDIC is the default value. ASCII should not be specified when EBCDIC is specified.

The ASCII translate table is in two parts. Normal ASCII is from X'00' to X'7F', but some users also have an "abnormal" ASCII that is a duplicate at displacements X'80' to X'FF'. COMPAREX supports both in the same table.

If ASCII translation is being used, processing parameter message CPX08I will specify ASCII.

CASE

Keyword Format

```
CASE={ UPPER }  
      { LOWER }  
      { MIXED }
```

specifies the translation of lower case characters, either EBCDIC or ASCII.

The default, CASE=UPPER sets the translation of lower case characters to periods. CASE=LOWER is logically equivalent to CASE=MIXED which sets the translation of lower case characters into themselves. If an option to TEXT such as TEXT= PANEL is specified, CASE=MIXED is set automatically.

As each line of SYSIN is read, it is automatically translated to upper case for scanning of legitimate keywords unless CASE=LOWER is specified.

Keyword Examples

```
CASE=(UPPER)  
CASE=MIXED  
CASE=lower          /* Same as CASE=mixed
```

DASH

Keyword Format

DASH={ C '-' }
 { t 'vv' }

specifies the character used as the underscore for differing bytes on the Differences Report.

The DASH character also is used as the character to separate a block of records from file SYSUT1 from a block of records from file SYSUT2 when FRAME is specified with TEXT processing.

When an inequality is discovered between two synchronized DATA records, the bytes that are different are underscored with the DASH character. COMPAREX prints the word 'DIFFERENCE' on the right side of the report, next to the record from file SYSUT2, on the same line as the DASH characters.

The default value is a dash. The user may specify any other character or hexadecimal value. If character data is requested, only one value is used (such as C'?'); if hexadecimal data is requested, two values are used (such as X'4F'). The DASH character being used is specified by processing parameter message CPX11I. In addition, the DASH character is shown immediately to the left of the word 'DIFFERENCE.' At the end of processing, COMPAREX shows a count of bytes underscored with message CPX74I.

The differing bytes are underscored with the DASH character on both the left-hand (hexadecimal) portion of the report and the right-hand (alphanumeric) portion of the report.

COMPAREX will accept any value given; the user needs only to be certain the value is a printable character on the printer's print train.

Keyword Examples

DASH=C'?'
DASH=X'EA'

DECIMAL

causes each line's relative displacement to be shown in decimal format.

DECIMAL and HEX are mutually exclusive. DECIMAL is the default value. If DECIMAL is specified, HEX should not be specified.

Each line's relative displacement is shown in the left-most column of the Differences Report.

If DECIMAL displacement is being used, processing parameter message CPX08I will specify DECIMAL.

An example of DECIMAL format is shown in "FORMAT=02,DECIMAL,NIBBLE (Excerpts)" on page 135. Note that the first line of the record includes bytes 1 through 32, the second line of the record includes bytes 33 through 64, the third line of the record includes bytes 65 through 96, and the fourth line of the record includes bytes 97 through the end of the record. This example shows DECIMAL with displacement relative to one (*the default*).

EBCDIC

specifies that the input files are in EBCDIC format. Unusual special characters are translated to periods after the comparison routine but before their printing on the Differences Report. EBCDIC stands for Extended Binary Coded Decimal Interchange Code.

EBCDIC and ASCII are mutually exclusive. EBCDIC should not be specified when ASCII is specified. EBCDIC is the default value.

If EBCDIC translation is being used, processing parameter message CPX08I will specify EBCDIC.

Note: The EBCDIC translate table may be customized at your shop by replacing CSECT COMPAREE in load module COMPAREX. See your installation guide for details.

FLDSONLY

specifies that only differing bytes defined by FIELD statements are underscored with the DASH character.

If the user specifies at least one FIELD1/FIELD2 combination, COMPAREX turns on the FLDSONLY indication automatically.

If FLDSONLY is being used, processing parameter message CPX11I will specify FLDSONLY.

```
CPX51I - RECORD NUMBER 4 ON FILE SYSUT1
1  F0F0F0F4 F5F7F0F6 00000000 0045706C 0000B28A C4C1E3C1 ... *00045706.....%....DATA-SYSUT1 *   O N E
33  F0F1F2F3 F4F50000 00000012 345C0000 30394040 40E3C8C9 ... *012345.....*.... THIS IS THE*   O N E
65  40C4C1E3 C140C6D6 D940E2E8 E2E4E3F1 40606060 60606040 ... * DATA FOR SYSUT1 ----- *   O N E
97  40404040 40404040 40404040 40404040 40404040 40404040 ... *                                     *   O N E

CPX52I - RECORD NUMBER 4 ON FILE SYSUT2
1  F0F0F0F4 F5F7F0F6 00000000 0045706F 0000B28A C4C1E3C1 ... *00045706.....?....DATA-SYSUT2 *   T W O
                                     -DIFFERENCE+
33  40C4C1E3 C140C6D6 D940E2E8 E2E4E3F2 40606060 60606040 ... * DATA FOR SYSUT2 ----- *   T W O
                                     -DIFFERENCE+
97  40404040 40404040 40404040 40404040 40404040 40404040 ... *                                     *   T W O
                                     ++++++++ -DIFFERENCE+
129 4040                                     *                                     *   T W O
    ****                                     **                                     -DIFFERENCE+
```

Figure 61. FORMAT=02,DECIMAL,NIBBLE (Excerpts)

FORMAT

Keyword Format

FORMAT=xy
(or F)

specifies the DATA formatting characteristics in how differences are displayed. The default is FORMAT=02, which implies the IBM dump format, full display of a SYSUT1 record followed by the differing lines of SYSUT2 with the differences underscored.

The possible values for x and their meaning are:

- 0 - Generates LINE = (32,HORIZONTAL). See "FORMAT=02,DECIMAL,NIBBLE (Excerpts)" on page 135 for an example of the IBM dump format.
- 1 - Generates LINE=(100,ALPHA). See "Alphanumeric Format - FORMAT=1y" on page 138 for an example of the alphanumeric format with 100 bytes per line.
- 2 - Generates LINE=(100,VERTICAL). See "PLUS and FORMAT=21 (Ditto Format)" on page 146 for an example of the "DITTO" (vertical hex) format with 100 bytes per line.

For the following discussion on the values of 'y', imagine a SYSUT1 record with every byte having a value of display character 'A' (C'A' or X'C1') and a SYSUT2 record similar except that byte number ten (relative to one) has a value of 'B' (C'B' or X'C2'). For example:

```
<----- SYSUT1 Record ----->
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA ONE
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA etc. ONE

<----- SYSUT2 Record ----->
AAAAAAAAABAAAAAAAAAAAAAAAAAAAAAAAAAAAA TWO
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA etc. TWO
```

The values for 'y' are independent of 'x', but their possible values are:

- 1 - Full display of SYSUT1 record followed by full display of SYSUT2 record with differences underscored. e.g.

```
CPX51I - RECORD NUMBER 1 ON FILE SYSUT1
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA ONE
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA etc. ONE

CPX52I - RECORD NUMBER 1 ON FILE SYSUT2
AAAAAAAAABAAAAAAAAAAAAAAAAAAAAAAAAAAAA TWO
- DIFFERENCE+
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA etc. TWO
```

- 2 - Full display of SYSUT1 record followed by only the differing lines of SYSUT2 with differences underscored. e.g.

```
CPX51I - RECORD NUMBER 1 ON FILE SYSUT1
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA ONE
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA etc. ONE

CPX52I - RECORD NUMBER 1 ON FILE SYSUT2
AAAAAAAAABAAAAAAAAAAAAAAAAAAAAAAAAAAAA TWO
- DIFFERENCE+
```

- 3 - Differing lines of SYSUT1 record followed by the differing lines of SYSUT2 with differences underscored. e.g.


```

CPX511 - RECORD NUMBER 1 ON FILE SYSUT1
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA   O N E
CPX521 - RECORD NUMBER 1 ON FILE SYSUT2
AAAAAAAAABAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA   T W O
- DIFFERENCE+

```

4 - Full display of SYSUT1 record interleaved with full display of SYSUT2 record and the differences underscored. e.g.

```

CPX511 - RECORD NUMBER 1 ON FILE SYSUT1
CPX521 - RECORD NUMBER 1 ON FILE SYSUT2

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA   O N E
AAAAAAAAABAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA   T W O
- DIFFERENCE+

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA etc.   O N E
AAAAAAAAABAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA etc.   T W O

```

5 - Full display of SYSUT1 record interleaved with only the differing lines of SYSUT2 and the differences underscored. e.g.

```

CPX511 - RECORD NUMBER 1 ON FILE SYSUT1
CPX521 - RECORD NUMBER 1 ON FILE SYSUT2

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA   O N E
AAAAAAAAABAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA   T W O
- DIFFERENCE+

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA etc.   O N E

```

6 - Differing lines of SYSUT1 record interleaved with differing lines of SYSUT2 and the differences underscored. e.g.

```

CPX511 - RECORD NUMBER 1 ON FILE SYSUT1
CPX521 - RECORD NUMBER 1 ON FILE SYSUT2

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA   O N E
AAAAAAAAABAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA   T W O
- DIFFERENCE+

```

All combinations of x and y are acceptable unless FIELD1/FIELD2 offsets are specified. In this case, regression to FORMAT=x1 or FORMAT=x2 may be enforced without expressed notification via a CPXnnA message. If differing FIELD1/FIELD2 offsets are specified such as:

FIELD1=(31 , 4 , Z) , FIELD2=(32 , 3 , P)

then

FORMAT=x3 , FORMAT=x5 , and FORMAT=x6

revert to

FORMAT=x2

and

FORMAT=x4

reverts to

FORMAT=x1 .

Keyword Examples

```
FORMAT=26  
F=3  
FORMAT=11
```

```
1  EXAMPLE OF FORMAT=1y, 100 BYTES PER LINE, UNPRINTABLE CHARACTERS  
101 WILL APPEAR AS ... PERIODS, lower case only if CASE=MIXED.
```

Figure 62. Alphanumeric Format - FORMAT=1y

GENFLDS

specifies that COMPAREX is to generate a visual representation of all IDENTITY, FIELD, and MASK statements on the Differences Report.

If GENFLDS is being used, processing parameter message CPX13I will be shown.

A page eject is done before and after each display so the user can separate the GENFLDS pages for reference as the Differences Report is studied. A clear plastic overlay could be made on a copying machine to aid in the analysis of the report.

COMPAREX uses the value of LINE specified with processing parameter message CPX08I to generate the GENFLDS representation. If this way, the GENFLDS representation is in the same format as the associated Differences Report.

HALT

Keyword Format

```
HALT={ NO }  
      { YES }  
      { COND }
```

specifies whether or not COMPAREX is to continue processing after extracting keywords. The default is HALT=NO which implies that in all cases, COMPAREX is to read files, compare etc regardless of any syntax errors the user may have committed. HALT=YES forces COMPAREX to terminate with message CPX31A after all keywords have been exhausted regardless of any syntax errors the user may have committed.

HALT=COND implies a conditional termination if the user committed any syntax errors within keywords. As delivered, HALT=COND is set in the installation defaults. From experience, most shops leave this in as the default.

Syntax errors are defined here as anytime COMPAREX underscores, with the latest DASH definition, non-blank characters that are not recognizable keywords. The literal "ERROR?" also is displayed to the right of the underscores alerting the user to the problem.

HELP

Keyword Format

```
HELP[={ALL}      ]
      [ {DATA}    ]
      [ {DISPLAY} ]
      [ {INPUT}   ]
      [ {OUTPUT}  ]
      [ {TEXT}    ]
```

specifies that COMPAREX is to generate a listing of valid COMPAREX keywords, along with a short description of the use of each keyword, on SYSPRINT, following message CPX011.

In this way, the user can obtain an on-line COMPAREX reference card. The HELP keyword is informational only and causes no other change to processing.

No option to HELP need be specified. If no option is specified or if HELP=ALL is specified, COMPAREX lists all valid keywords.

If HELP=DATA is specified, COMPAREX lists only the Data Files Synchronization keywords (see "Chapter 6 - Data File Synchronization Keywords" on page 99).

If HELP=DISPLAY is specified, COMPAREX lists only the Display Processing keywords.

If HELP=INPUT is specified, COMPAREX lists only the Input Processing keywords (see "Chapter 5 - Input Processing Keywords" on page 69).

If HELP=OUTPUT is specified, COMPAREX lists only the Output Processing keywords (see "Chapter 8 - Output Processing Keywords" on page 124).

If HELP=TEXT is specified, COMPAREX lists only the Text File keywords (see "Chapter 7 - TEXT Keywords" on page 112).

Keyword Format

```
HELP
HELP=TEXT
HELP=(DATA)
```

When HELP or HELP=ALL is specified, a printout of the general syntax for each keyword is generated. For example:

```
CPX011 - INPUT PROCESSING KEYWORDS
CONTINUE - STOP DISPLAYING DIFFERENCES BUT CONTINUE BEYOND MAXDIFF
CPXIFACE=XXX - LOAD MODULE NAME FOR COMPAREX INTERFACE EXIT
DATA|CSECT| - FILES HAVE INTER-RECORD RELATIONSHIP
DESEN=(DDD,T'VVVV') - DESENSITIZE BOTH FILES
DESEN1=(DDD,T'VVVV') - DESENSITIZE SYSUT1
DESEN2=(DDD,T'VVVV') - DESENSITIZE SYSUT2
DIRECTORY=|USER/SPF/PDF| - PDS DIRECTORY PROCESSING ONLY
FIELD=(D,L,|C/Z/P/B|) - FIELD COMPARISON BOTH FILES (MAX 400)
FIELD1=(D,L,|C/Z/P/B|) - FIELD COMPARISON FOR SYSUT1
FIELD2=(D,L,|C/Z/P/B|) - SYSUT2 FILE FIELD OFFSET (SEE FIELD1)
FILTERIN=(|MEMBER,/M,|D1-D2,OP,T'VVVVV') - INCLUSIVE FILTER USING 'AND' LOGIC
FILTEROUT=(|MEMBER,/M,|D1-D2,OP,T'VVVVV') - EXCLUSIVE FILTER USING 'AND' LOGIC
FILTORIN=(|MEMBER,/M,|D1-D2,OP,T'VVVVV') - INCLUSIVE FILTER USING 'OR' LOGIC
FILTOROUT=(|MEMBER,/M,|D1-D2,OP,T'VVVVV') - EXCLUSIVE FILTER USING 'OR' LOGIC
IDENTITY=(DDD,OP,T'VVVVV') - FIELD COMPARISONS BY RECORD ID
IGNORSIN - IGNORE SIGN DIFFERENCES
MASK=(DDD,LLL) - MASKING AREAS FOR BOTH FILES
MASK1=(DDD,LLL) - MASKING AREAS FOR SYSUT1
MASK2=DDD - SYSUT2 FILE MASK OFFSET (SEE MASK1)
MODE=|SYSTEMS/APPLICATIONS| - USER ORIENTATION
SKIPUT1=NNN - SKIP OVER FIRST NNN RECORDS OF FILE SYSUT1
SKIPUT2=NNN - SKIP OVER FIRST NNN RECORDS OF FILE SYSUT2
STOPAFT=NNN - STOP AFTER NNN RECORDS READ ON SYSUT1 OR SYSUT2
SYSUT1=(|PAN/LIB/OTH|,MEMBER=XXX,INCLUDE=|NO/YES|,LEVEL=N,PARM=XXX)
SYSUT1=|QSAM/ISAM/VSAM/PDS| - EXPLICIT SYSUT1 ACCESS METHOD, NOT RECOM.
```

SYSUT2 IS SAME AS SYSUT1
WILDCARD=T'VV' - GENERIC CHARACTER

CPX011 - OUTPUT PROCESSING KEYWORDS
COPYDIFF=(|PAN/LIB/GEM/IEBUPDTE/MEMBER/OTH|,VERS=|YES/NO|,PASS=|YES/NO|,STAMP=|NO/YES|,SEQFLD=738)
INSERT=|++C/-INS/-INS| - TEXT/COPYDIFF FORMATTING
DELETE=|++C/-DEL/-DEL| - TEXT/COPYDIFF FORMATTING
REPLACE=|++C/-REP/-REP| - TEXT/COPYDIFF FORMATTING
SYSUT3=|QSAM/ISAM/VSAM| - EXPLICIT SYSUT3 ACCESS METHOD

CPX011 - DATA FILE SYNCHRONIZATION KEYWORDS
KEY=(D,L,|C/Z/P/B|,|A/D/R|) - KEY SYNCH OF BOTH FILES (MAX 40)
KEY1=(D,L,|C/Z/P/B|,|A/D/R|) - KEY SYNCH OF SYSUT1 FILE
KEY2=(D,L,|C/Z/P/B|,|A/D/R|) - KEY SYNCH OF SYSUT2 (SEE KEY1)
SEGMENT=(D1,EQ,T'VVVVV',(|A/D/R|,D2,LLL)) - DATA BASE SEGMENT PROCESSING

CPX011 - DISPLAY PROCESSING KEYWORDS
ASCII - ASCII TRANSLATE TABLE
CASE=|UPPER/LOWER/MIXED| - TRANSLATION OF LOWER CASE CHARACTERS
DASH=T'VV' - DEFINITION OF DIFFERENCES UNDERSCORE
DECIMAL - RELATIVE DISPLACEMENTS IN DECIMAL
EBCDIC - EBCDIC TRANSLATE TABLE
FLDSONLY - DIFFERENCES UNDERSCORED ONLY ON SPECIFIED FIELDS
FORMAT=XY - X=|0/1/2|, Y=|1/2/3/4/5/6| - DATA FORMATTING
GENFLDS - VISUAL INTERPRETATION OF FIELDS TO BE COMPARED
HALT=|YES/NO/COND| - HALT EXECUTION ON SYSIN ERROR
HEX - RELATIVE DISPLACEMENTS IN HEX
INTERLEAVE=NNN - INTERLEAVE DIFFERING PRINTED LINES
KEYONLY - DISPLAY LINES CONTAINING CONTROL FIELDS ON SYNCH MISMATCHES
LINE=(NNN,|HORIZONTAL/ALPHA/VERTICAL|) - PRINT LINE CHARACTER REPRESENTATION
MAXDIFF=NNN - STOP OR PAUSE (SEE CONTINUE) AFTER NNN DIFFERING RECORDS
MBRHDR=|YES/NO/COND/MATCH| - HEADINGS ON MEMBERS
NIBBLE - UNDERSCORE HALF-BYTE NIBBLES IF DUMP FORMAT
PAGE=NNN - NNN AT LEAST 10 (DEFAULT IS 58) LINES PER PAGE
PLUS=T'VV' - DEFINITION OF EXTRA BYTES UNDERSCORE
PRINT=|MATCH/NOMATCH/MISMATCH/NOMISMATCH| - PRINT CONTROL

CPX011 - TEXT FILE KEYWORDS
BUFF=NNN - BUFFER SIZE
FRAME=|NUM/YES/NO| - SURROUND BLOCKS OF DIFFERING RECORDS
MLC=NNN - MATCHING LINE COUNT
SQUEEZE=T'VV' - CHARS TO BE SQUEEZED OUT
TEXT=|COBOL/\$COBOL/.JCL/BAL/CLIST/REPORT ETC.| TEXT COMPARE
PRINT=FULL - ALL OF SYSUT1 WITH DIFFERENCES DISPLAYED IN CONTEXT

HEX

causes each line's relative displacement to be shown in hexadecimal format. DECIMAL and HEX are mutually exclusive. DECIMAL is the default. If HEX is specified, DECIMAL should not be specified. Each line's relative displacement is shown in the left-most column of the Differences Report. If HEX displacement is being used, processing parameter message CPX08I will specify HEX. If MODE=SYSTEMS is specified and DECIMAL is not specified, COMPAREX turns on HEX automatically.

IGNORSIN

specifies that DATA differences in packed fields with unlike signs are to be ignored. If one record contains packed fields with signs of C and the other record contains packed fields with signs of E, they can be ignored. This keyword causes COMPAREX to scan every byte of both synchronized records for packed fields and make them signs of E before comparison begins. When the two records are compared, these sign differences are effectively ignored. Certain restrictions apply to the use of IGNORSIN. They are
No FIELD's or MASK's are allowed.
Crippled for TEXT processing.
Not applicable for CSECT Parsing

Keyword Examples

IGNORSIN
IGNORSIN=YES

INTERLEAVE

Keyword Format

INTERLEAVE=nn
(or ILV)

specifies the number of lines that are blocked together from a SYSUT1 record before displaying a similar number of lines from a SYSUT2 record. This only has meaning in DATA logic when the value of y in FORMAT=xy is 4, 5, or 6. If entered, the value of nn must be at least 1. If not entered, the default value is 1. It is left as an exercise for the user to see that specifying a very high value such as INTERLEAVE=10000 with FORMAT=04 is logically equivalent to specifying FORMAT=01 and disregarding INTERLEAVE.

Keyword Examples

INTERLEAVE=1
ILV=010

KEYSONLY

specifies that when synchronization mismatches, either from KEY or SEGMENT synchronization, occur, only the LINES that completely contain any control fields will be displayed. This is particularly useful when comparing DATA files with relatively large records (500 bytes or more), and the user is not concerned with flipping through the pages of the display for inserted and deleted records. Another possibility is to specify PRINT=NOMISMATCH but that ignores the inserted/deleted records completely from the Differences Report.

KILLRC

specifies that whatever return code is going to be sent back to the operating system, it is overridden to be zero. This keyword is rarely used.

Keyword Examples

KILLRC
KILLRC=NO

LINE

Keyword Format

```
LINE=( { 32 } [ { , HORIZONTAL } ] )  
      { nn } [ { , HOR } ]  
            [ { , ALPHA } ]  
            [ { , VERTICAL } ]  
            [ { , VER } ]
```

specifies the number of bytes displayed on each line and the method for that display. The default value for nn is 32 but may range from 10 to 110. If HORIZONTAL (or HOR) is specified, the value of nn must be 32. All other combinations are valid. LINE=(32,HORIZONTAL) generates IBM dump format. LINE=(nn,ALPHA) generates an alphanumeric line of length nn. LINE=(nn,VERTICAL) generates the "DITTO" format with line length nn.

If both LINE and FORMAT are specified, the LINE specification overrides the x in FORMAT=xy accordingly.
See "PLUS and FORMAT=21 (Ditto Format)" on page 146 for an example of the Ditto format with LINE=(100,VERTICAL).

Keyword Examples

```
LINE=80  
LINE=(32,HOR)  
LINE=(110,VERTICAL)
```

MAXDIFF

Keyword Format

```
MAXDIFF={999999999999}  
        {nn}
```

specifies the maximum number of differences to be displayed on the Differences Report. A difference can be two matched records with differing data or one record that is not matched on the other file (any inserted record under KEY or SEGMENT synchronization or any extra record on the end of the longer file).

The MAXDIFF value being used is specified by processing parameter message CPX04I. When COMPAREX has displayed the specified number of differences, the utility will issue message CPX67I. At that time, if CONTINUE is not specified, COMPAREX will execute its end-of-job routines; if CONTINUE is specified, COMPAREX will read and compare records, adding to processing totals but not printing records on the Differences Report.

When comparing partitioned datasets (PDS's) and DIRECTORY is not specified, differing directory records do not count toward the MAXDIFF limit; only differing member records are counted toward the MAXDIFF limit.

The user should ALWAYS give a MAXDIFF specification to prevent large printouts when errors occur.

Keyword Examples

```
MAXDIFF=10  
MAXDIFF=(999)
```

MBRHDR

Keyword Format

```
MBRHDR={YES}  
        {COND}  
        {NO}  
        {MATCH}
```

specifies whether or not COMPAREX is to display a member header for each member of a directory-embedded dataset compare. **MBRHDR=YES**, the default, forces a page break and heading to be printed for each member regardless of whether or not there are any differences.

MBRHDR=COND specifies that a page break and member header are to be issued only if there is indeed at least one difference in the two members that synchronized together or a member insertion. When large libraries are compared, and only a few members differ, this abbreviates the Differences Report considerably.

MBRHDR=NO forces all page breaks off and all Differences Reports by member suppressed. Only statistics by member are gathered as to how many members differed that synchronized together and how many inserted members were on each file. Using this option speeds up the overall comparison considerably when large libraries are involved. When two members synchronize together because their names match, the comparison is abruptly terminated at the first difference. All detail is lost, however, as to which members differed and where.

MBRHDR=MATCH is similar to **MBRHDR=COND** except that inserted members are not displayed. Only member names that Match and have differences warrant a page break.

Note: Statistics by member are gathered and displayed in all three cases with message CPX78I.

For example, if two directory-embedded datasets contain members:

-SYSUT1-	-SYSUT2-
MEMBER10	MEMBER10 (identical)
MEMBER20	MEMBER15
MEMBER30	MEMBER20 (different)
MEMBER40	MEMBER40 (identical)
MEMBER50	

With **MBRHDR=COND**, the difference report looks like:

```
MEMBER15          D I F T W O  2

SYSUT1=DSNUT1(MEMBER20),SYSUT2=DSNUT2(MEMBER20)
{List of differences within the member}

MEMBER30          D I F O N E  3

MEMBER50          D I F O N E  5
```

With **MBRHDR=MATCH**, the difference report looks like:

```
SYSUT1=DSNUT1(MEMBER20),SYSUT2=DSNUT2(MEMBER20)
{List of differences within the member}
```

NIBBLE

specifies that each half byte is to be compared and, if different, underscored with the DASH character.

For the **NIBBLE** keyword to be used, **LINE** must be set (or defaulted) to "(32,HORIZONTAL)".

If **NIBBLE** is being used, processing parameter message CPX11I will specify **NIBBLE**. **NIBBLE** may be used only with **DATA** comparison logic; no bytes are underscored with **TEXT** comparison logic.

"FORMAT=02,DECIMAL,NIBBLE (Excerpts)" on page 135 shows an example of **NIBBLE** on the Differences Report.

PAGE

Keyword Format

PAGE={ 58 }
 { nn }

specifies the number of print lines on each page. The value may be between 10 and 99999999. The default value is 58.

The PAGE value being used is specified by processing parameter message CPX08I.

This number sets the maximum number of lines to be printed on a page. A low value for PAGE, such as 10, will cause more pages to be printed since COMPAREX will advance to the top of the page and will write the two heading lines (showing time, date, page number, and input file dataset names) each time that number of lines has been written. A high value for PAGE, such as 999999, will cause COMPAREX to print over the fanfold page boundaries, perhaps saving some paper.

Keyword Examples

PAGE=76
PAGE=(999)

PLUS

Keyword Format

PLUS={ C '+' }
 { t 'vv' }

specifies the character used as the underscore on the Differences Report for excess bytes when the record from SYSUT2 is longer than the record from SYSUT1.

The PLUS character also is used as the surrounding character when FRAME is specified with TEXT processing.

The PLUS value being used is specified by processing parameter message CPX11I. In addition, the PLUS character is shown immediately to the right of the word 'DIFFERENCE.' At the end of processing, COMPAREX shows a count of excess bytes underscored as the second figure in message CPX74I.

The default value is a plus character. The user may specify any other character or hexadecimal value. If character data is requested, only one value is used (such as C'*'); if hexadecimal data is requested, two values are used (such as X'4F').

Excess bytes are not underscored with TEXT processing logic.

The excess bytes are underscored with the PLUS character on both the left-hand (hexadecimal) portion of the report and the right-hand (alphanumeric) portion of the report.

COMPAREX will accept any value given; the user needs only to be certain the value is a printable character on the printer's print train.

"PLUS and FORMAT=21 (Ditto Format)" on page 146 illustrates the use of PLUS=C' on the Differences Report.

Keyword Examples

PLUS=C '*'
PLUS=X'5C'

```
DSPL  |...+....1....+....2....+....3....+....4....+....5....+
CPX51I - RECORD NUMBER 1 ON FILE SYSUT1
1      00045678.....>DATA-SYSUT1 012345.....*..... TH
      FFFFFFFF000004680086CCEC6EEEEEF4FFFFFFF000001350033444EC
      000456780000057C002E4131028243100123450000024C000900038
101
      444444444444444444444444
      000000000000000000000000
CPX52I - RECORD NUMBER 1 ON FILE SYSUT2
1      00045688.....DATA-SYSUT2 012345.....*..... TH
      FFFFFFFF000004680087CCEC6EEEEEF4FFFFFFF000001350033444EC
      000456880000058C00284131028243200123450000024C000900038
101
      4444444444444444444444444444444444444444444444444444444
      0000000000000000000000000000000000000000000000000000000
      ))))))) -DIFFERENCE)
```

Figure 63. PLUS and FORMAT=21 (Ditto Format)

PRINT

Keyword Format

```
PRINT={ MATCH }  
      { NOMATCH }  
      { MISMATCH }  
      { NOMISMATCH }  
      { FULL }
```

specifies which synchronized MATCH's or MISMATCH's will be printed.

This form of the PRINT keyword may be used with DATA comparison logic or with DIRECTORY (which is a variation on DATA logic). The defaults are MATCH and MISMATCH. If MATCH is specified, NOMATCH should not be specified; if MISMATCH is specified, NOMISMATCH should not be specified. MATCH and NOMATCH are mutually exclusive; MISMATCH and NOMISMATCH are mutually exclusive.

The PRINT parameters being used are specified by processing parameter message CPX05I.

Options:

MATCH - specifies that if records synchronize together and they do not compare exactly, COMPAREX will print both records on the Differences Report, underscoring the differing bytes.

NOMATCH - specifies that if records synchronize together, COMPAREX will print neither record on the Differences Report.

MISMATCH - specifies that if an out-of-synchronization situation occurs, COMPAREX will print this record on the Differences Report.

NOMISMATCH - specifies that if an out-of-synchronization situation occurs, COMPAREX will not print this record on the Differences Report.

FULL - specifies that all records from SYSUT1 will be printed in context with the differing records. This is not applicable to Random keys or Csect Parsing.

Keyword Examples

```
PRINT=NOMATCH  
PRINT=(MATCH,NOMISMATCH)  
PRINT=FULL
```

Chapter 10 - Messages

Throughout the COMPAREX job, the utility prints messages on SYSPRINT to show the user the results of the processing. Messages show the defaults COMPAREX will use, the keywords entered by the user that modify those defaults, and the source of each record shown on the Differences Report.

In addition, COMPAREX uses its messages to notify the user of actions to take when errors occur. These errors can be either in the processing environment or in the set of user-entered keywords.

Finally, COMPAREX uses messages to show processing totals at the end of each job. These messages show tallies of input records as well as the numbers of differences found.

Messages Issued During Job Initialization

Messages issued by COMPAREX at the beginning of each job are used to show the parameters in effect for the run. For the most part, these messages are for the information of the user; no action, other than a review, is usually needed. The only messages at job initialization requiring user action are those describing files that cannot be opened or keywords that are unable to be interpreted.

CPX00I

Message Format

CPX00I - input line from SYSIN file

Each 80-byte line from the SYSIN file is shown to the right of this message.

If "ERROR?" is printed on the right of the report, on the line under the line with message number CPX00I, this is an ACTION message. If HALT=COND has been specified, then the utility will terminate with message CPX30A and return code 16 after issuing all informational messages but before reading any records from SYSUT1 or SYSUT2.

ACTION: Examine the line containing the literal "ERROR?" to find the underscores. Under the line containing the message number CPX00I, COMPAREX has underscored the characters that cannot be interpreted.

Change the specification. If a keyword is misspelled or if a keyword's parameters are incorrectly given, correct the specification. Refer to the description of the keyword in this manual for information about keyword parameters and their values. If the underscores identify notes or comments, precede the comments with a "/" (but not in columns 1 and 2) which delineates comments from keywords. Alternatively, specify HALT=NO, which forces COMPAREX to continue processing regardless of any syntax errors.

If there are characters on the line containing the message number CPX00I that are not underscored, COMPAREX has interpreted these characters as correct keywords, and the utility has modified its default processing with these keywords.

If "ERROR?" is not printed on the right of the report under the message line, this is an informational message.

To review: Examine the line to insure that the keywords and their parameters were correctly entered.

CPX01I

Message Format

CPX01I - (INPUT PROCESSING KEYWORDS)
(OUTPUT PROCESSING KEYWORDS)
(DATA FILE SYNCHRONIZATION KEYWORDS)
(DISPLAY PROCESSING KEYWORDS)
(TEXT FILE KEYWORDS)

This is an informational message. HELP has been specified. Refer to the description of the HELP keyword in "Chapter 9 - Display Processing Keywords" on page 130.

If the HELP keyword without any option was found or HELP=ALL has been specified, COMPAREX lists each valid keyword, along with a short description of the use of each keyword, on SYSPRINT, following message CPX01I.

If HELP=INPUT was specified, COMPAREX lists each input processing keyword, along with a short description of the use of each input processing keyword, on SYSPRINT, following message CPX01I.

If HELP=OUTPUT was specified, COMPAREX lists each output processing keyword, along with a short description of the use of each output processing keyword, on SYSPRINT, following message CPX01I.

If HELP=DATA was specified, COMPAREX lists each data file synchronization keyword, along with a short description of the use of each data file synchronization keyword, on SYSPRINT, following message CPX01I.

If HELP=DISPLAY was specified, COMPAREX lists each display processing keyword, along with a short description of the use of each display processing keyword, on SYSPRINT, following message CPX01I.

If HELP=TEXT was specified, COMPAREX lists each text file keyword, along with a short description of the use of each text file keyword, on SYSPRINT, following message CPX01I.

The HELP keyword causes no other change to COMPAREX's processing.

CPX02I

Message Format

CPX02A - SYSIN PARAMETER FILE MISSING OR INVALID, TAKING ALL INTERNAL DEFAULTS

If keywords were used for this run of COMPAREX, this is an ACTION message.

ACTION: Examine JCL to determine why COMPAREX was unable to open SYSIN.

Possible reason:

//SYSIN was missing or misspelled.

If keywords were not used for this run of COMPAREX, this is an informational message.

To review: No error correction is necessary. The COMPAREX run took all defaults.

CPX03I

Message Format

CPX03I - EXECUTION OF jobname.stepname.procstepname - VALUES EXTRACTED/DEFAULTED:

This is an informational message. It is printed after all keywords have been extracted, but before the files to be compared are opened. Following this message, COMPAREX prints the parameters it will use during the execution. These parameters are explicitly stated in the messages that immediately follow message CPX03I.

In addition, message CPX03I helps the user identify the step of the JCL by displaying jobname, stepname, and procstepname.

jobname: this name is taken from the JOB statement (such as //jobname JOB D,...).

stepname: this is the name of the step that actually invokes COMPAREX (for example, //stepname EXEC PGM=COMPAREX).

procstepname: if a procedure is called that invokes COMPAREX, that name is used. For example:

```
//COMPARE      PROC
//stepname     EXEC PGM=COMPAREX
//              PEND
//procstepname EXEC COMPARE
```

CPX04I

Message Format

CPX04I - MAXDIFF=n1[,CONTINUE],STOPAFT=n2

This is an informational message. It is printed immediately after message CPX03I. If MAXDIFF was specified, the input value for MAXDIFF is displayed instead of 'n1.' If MAXDIFF was not specified, MAXDIFF=9999999999 is displayed. This number is the maximum number of differences COMPAREX will print on the Differences Report during the run.

If CONTINUE was specified, the word CONTINUE is shown; otherwise, the word CONTINUE is not shown.

If STOPAFT was specified, the input value for STOPAFT is displayed instead of 'n2.' If STOPAFT was not specified, STOPAFT=9999999999 is displayed. This number is the maximum number of records COMPAREX will read from either input file (this number does not include records bypassed as a result of SKIPUT1 or SKIPUT2 keywords).

To review: Examine the counters displayed. Correct or change the keywords entered before any subsequent run, if desired.

CPX05I

Message Format

CPX05I - PRINT=(**MATCH**),(**MISMATCH**) [,**FULL**]),**MBRHDR**=(**YES**) ,**HALT**=(**NO**) [,**KEYSONLY**],**KILLRC**=(**NO**)
(**NOMATCH**) (**NOMISMATCH**) (**NO**) (**YES**) (**YES**) (**YES**)
(**COND**) (**COND**)

This is an informational message. If an option to PRINT was specified, this message will show the PRINT parameters from that keyword. The default is PRINT=(MATCH,MISMATCH).

If MBRHDR was specified, this message will show the MBRHDR parameter from that keyword. The default is MBRHDR = YES.

If HALT was specified, this message will show the HALT parameter from that keyword. The default is HALT = NO but it is recommended that the shop set HALT = COND as an Installation Default.

If KEYSONLY was specified, this message will show the KEYSONLY is turned on. The default is not KEYSONLY.

To review: Examine the parameters displayed. If necessary, check the description of the PRINT keyword in "Chapter 9 - Display Processing Keywords" on page 130.

CPX06I

Message Format

CPX06I - WILDCARD=(**C'.'**) ,**MODE**=(**APPLICATIONS**),(**ALL DISPLACEMENTS RELATIVE TO** (**ONE**))
(**t'vv'**) (**SYSTEMS**) (**ZERO**)

This is an informational message. It is shown on every COMPAREX run.

If at least one WILDCARD specification was found, the last WILDCARD value is shown instead of C'.'; if COMPAREX did not find a WILDCARD specification, WILDCARD=C'.' is shown.

If MODE=SYSTEMS was specified, and it was not followed by a MODE=APPLICATIONS keyword, message CPX06I shows MODE=SYSTEMS (ALL DISPLACEMENTS RELATIVE TO ZERO).

If MODE=SYSTEMS was not specified, message CPX06I shows MODE=APPLICATIONS (ALL DISPLACEMENTS RELATIVE TO ONE).

To review: Examine the parameters displayed to ensure that the correct WILDCARD value and MODE were used for the execution.

CPX07I

Message Format

```
CPX07I - SYNCHRONIZATION KEY(S):
        KEY=(ddd,[[[ [C,C] [C,A] ]
                [[C,Z] [C,D]
                [[C,P] [C,R]
                [[C,B]
KEY1=(ddd,[[[ [C,C] [C,A] ]),KEY2=([ddd[[[ [C,C] [C]
                [[C,Z] [C,D]                [C,Z]
                [[C,P] [C,R]                [C,P]
                [[C,B]                [C,B]
```

This is an informational message. It is shown when COMPAREX finds one or more KEY (or KEY1 and KEY2 pairs) specifications.

The KEYs are ordered in the same way they were specified. COMPAREX processes as if the first KEY shown under message CPX07I is the most major KEY and the last KEY shown is the most minor KEY.

If none of A-ascending, D-descending, or R-random was specified for a KEY, COMPAREX will assume A next to the KEY to show an ascending key.

If no exotic specification for type such as Z - Zoned; P - Packed; or B - Binary is made, the default of C - Character is assumed.

To review: Examine the parameters displayed to insure that the correct set of KEYs was used for the job.

CPX08I

Message Format

CPX08I - {DECIMAL},{EBCDIC},CASE={UPPER},LINE={n1,{HORIZONTAL}},PAGE=n2
(HEX) (ASCII) (MIXED) (ALPHA)
(VERTICAL)

This is an informational message. It is shown on every COMPAREX run.

If HEX was specified, and it was not followed by a DECIMAL keyword, the literal 'HEX' is shown and COMPAREX shows the relative displacement of each line of each record on the Differences Report in hexadecimal format; if MODE=SYSTEMS was specified and COMPAREX did not find a DECIMAL specification, the literal 'HEX' is shown and COMPAREX shows the relative displacement of each line of each record on the Differences Report in hexadecimal format; otherwise, the literal 'DECIMAL' is shown and COMPAREX shows the relative displacement of each line of each record on the Differences Report in decimal format.

The alphanumeric representation of the characters in the records on the Differences Report can be shown in either EBCDIC or ASCII format. If ASCII was specified, and it was not followed by an EBCDIC keyword, the literal 'ASCII' is shown and COMPAREX uses its ASCII translation table to translate each byte in each record on the Differences Report to an alphanumeric character for printing; otherwise, the literal 'EBCDIC' is shown and COMPAREX uses its EBCDIC translation table to translate each byte in each record on the Differences Report to an alphanumeric character for printing.

If CASE was specified, the proper option, either UPPER or MIXED (same as LOWER), is displayed here. In the absence of any CASE specification, the default is CASE=UPPER unless an option to TEXT, such as TEXT=PAGE, is made, in which case, CASE is set to MIXED.

If LINE was specified and/or FORMAT was specified, the composite results of those specifications (LINE takes precedence over FORMAT), are displayed here. However, if an option to TEXT, such as TEXT=COBOL, is made, all LINE and FORMAT specifications are ignored and LINE is set to (80,ALPHA) or (110,ALPHA) in the case of TEXT=REPORT.

Composite results are as follows: If

LINE=(77,VERTICAL),FORMAT=06

is specified, the result within message CPX08I is

LINE=(77,VERTICAL)

but within message CPX25I, the result will be

DATA,FORMAT=26,INTERLEAVE=1.

If PAGE was specified, the value on that keyword is shown instead of 'n2'; otherwise, the literal '58' is shown instead of 'n2.'

To review: Examine the parameters displayed to insure that the Differences Report was formatted as desired.

CPX09I

Message Format

CPX09I - SKIPUT1=n1,SKIPUT2=n2

This is an informational message. It is shown only when SKIPUT1 or SKIPUT2 has been specified.

If SKIPUT1 has been specified, the value on that keyword is shown instead of 'n1'; otherwise, the number zero is shown instead of 'n1.' COMPAREX will read (skip over) this number of records on SYSUT1 before passing any SYSUT1 record to the compare routines.

If SKIPUT2 has been specified, the value on that keyword is shown instead of 'n2'; otherwise, the number zero is shown instead of 'n2.' COMPAREX will read (skip over) this number of records on SYSUT2 before passing any SYSUT2 record to the compare routines.

To review: Examine the parameters displayed to insure that the correct number of records was bypassed (skipped over) on each input file.

CPX10I

Message Format

CPX10I - FILTERS:
FILTERIN=(MEMBER,/CSECT,1d1[-d2],op,t'vvvv')
FILTEROUT=(MEMBER,/CSECT,1d1[-d2],op,t'vvvv')
FILTORIN=(MEMBER,/CSECT,1d1[-d2],op,t'vvvv')
FILTOROUT=(MEMBER,/CSECT,1d1[-d2],op,t'vvvv')

This is an informational message. It is shown only when COMPAREX finds one or more filtering specifications.

The filters are ordered in the same way they were specified. COMPAREX processes the filters in the order shown under message CPX10I.

COMPAREX converts the last filtering keyword of each type (record and member) to an 'AND' logic filter.

To review: Examine the statements displayed to insure that the correct set of filters was used for the job. If necessary, check the descriptions of the FILTERIN, FILTEROUT, FILTORIN, and FILTOROUT keywords in "Chapter 5 - Input Processing Keywords" on page 69.

CPX11I

Message Format

CPX11I · DASH=(C'-') ,PLUS=(C'+') [,NIBBLE] [,FLDONLY]
(t'v1') (t'v2')

This is an informational message. It is shown on every COMPAREX run.

If DASH was specified, the value on that keyword is shown instead of 't'v1'; otherwise, the literal 'DASH=C'-' is shown. COMPAREX will use this character to underscore differing bytes on the Differences Report.

If PLUS was specified, the value on that keyword is shown instead of 't'v2'; otherwise, the literal 'PLUS=C'+ ' is shown. COMPAREX will use this character to underscore excess bytes on the Differences Report when a SYSUT2 record is longer than a paired SYSUT1 record.

If NIBBLE was specified, the word NIBBLE is shown and only differing half-bytes will be underscored on the Differences Report.

If FLDONLY was specified, or if the user entered one or more sets of FIELD1 and FIELD2 keywords, the word FLDONLY is shown and only differing bytes defined by FIELD statements are underscored with the DASH character.

To review: Examine the parameters displayed to insure that the keywords were correctly entered.

CPX12I

Message Format

CPX12I · IDENTITIES, DESENSITIZING, FIELDS, AND MASKS:
FIELD=(dd, lll, t) 1 FIELD=(dd, lll, t)
IDENTITY=(dd, op, t'vv') 2 IDENTITY=(dd, op, t'vv')
MASK1=(dd, ll), MASK2=(dd, ll) 3 FIELD=(dd, lll)
MASK=(dd, lll) 4 FIELD=(dd, lll)
FIELD1=(d, l), FIELD2=d 5 FIELD1=(d, l), FIELD2=d
DESEN=(dd, t'vv') 6 DESEN=(dd, t'vv')
DESEN1=(dd, t'vv') 7 DESEN1=(dd, t'vv')
DESEN2=(dd, t'vv') 8 DESEN2=(dd, t'vv')

This is an informational message. It is shown only when COMPAREX finds one or more IDENTITY, FIELD (or FIELD1 and FIELD2 pair), MASK (or MASK1 and MASK2 pair), or DESEN (DESEN1 or DESEN2 also) keywords as specifications.

The message has three parts. The left part is the same as specified, the middle part is a sequence number, and the right part is COMPAREX's interpretation of the input for processing. If no MASK (or MASK1 and MASK2) specifications are made, the right part is not shown.

COMPAREX always inserts a final IDENTITY test for records which do not pass any of the user's IDENTITY tests. This IDENTITY and its associated FIELD are shown in this way:

IDENTITY=(CATCH-ALL)
FIELD=(1, END)

To review: Examine the left part of the message to insure that the correct set of IDENTITIES, FIELDS, MASKS, and DESENSs was used for the job. Check the order of the keywords if IDENTITIES are used.

The middle part is a sequence number assigned to each IDENTITY and FIELD. This sequence number is used to refer to these same IDENTITIES and FIELDS on the GENFLDS output and on message CPX52I.

The right part is shown if any any MASK (or MASK1 and MASK2 pair) specifications were made. COMPAREX evaluates the MASKS and creates FIELDS. For example:

MASK=(6 , 9)

would generate two FIELDS:

FIELD=(1 , 5 , C)
FIELD=(15 , END)

and these two FIELDS would be shown as the right part of message CPX12I.

Here is a more complete example. If the specifications are:

```
FIELD=(4,009)
ID=(004,LE,C'ABC')
FIELD=(15,END),MASK=(24,3),MASK=(20,1)
IDENTITY=(4,EQ,C'XYZ')
FIELD1=(15,8,Z),FIELD2=(17,5,P)
MASK1=(21,4,B),MASK2=23
MASK=(1,5),MASK=(60,END)
IDENTITY=(7,EQ,X'EF'),MASK=(25,4)
```

Then, message CPX12I will show:

FIELD=(4,9,C)	1	FIELD=(4,9,C)
IDENTITY=(4,LE,C'ABC')	2	IDENTITY=(4,LE,C'ABC')
FIELD=(15,END)	3	FIELD=(15,5,C)
MASK=(24,3,C)	4	FIELD=(27,END)
MASK=(20,1,C)	5	FIELD=(21,3,C)
IDENTITY=(4,EQ,C'XYZ')	6	IDENTITY=(4,EQ,C'XYZ')
FIELD1=(15,8,Z),FIELD2=(17,5,P)	7	FIELD1=(15,8,Z),FIELD2=(17,5,P)
MASK1=(21,4,B),MASK2=23		
MASK=(1,5)		
MASK=(60,END)		
IDENTITY=(7,EQ,X'EF')	8	IDENTITY=(7,EQ,X'EF')
MASK=(25,4,C)	9	FIELD=(1,24,C)
	10	FIELD=(29,END)
IDENTITY=(CATCH-ALL)	11	IDENTITY=(CATCH-ALL)
FIELD=(1,END)	12	FIELD=(1,END)

Refer to "IDENTITY, FIELD, MASK, and DESEN Messages" on page 91 for an actual example.

CPX13I

Message Format

CPX13I - GENFLDS

This is an informational message. It is shown only when COMPAREX finds the GENFLDS keyword and one or more IDENTITY or FIELD (or FIELD1 and FIELD2 pair) specifications.

Immediately after the CPX13I message, COMPAREX advances to the top of the page to create the first GENFLDS visual representation, using the line length specified by the value of LINE given in the CPX08I message.

To review: Examine the generated visual representations. Modify IDENTITY, FIELD, and MASK keywords as necessary to correctly describe the records. See the description of the GENFLDS keyword in "Chapter 9 - Display Processing Keywords" on page 130.

CPX15I

Message Format

```
CPX15I - COPYDIFF[=(PAN ,STAMP=NO ,VERS=YES,PASS=YES)]  
      [ LIB          YES      NO      NO      ]  
      [ GEM          ]  
      [ IEBUPDTE ,SEQFLD='ddl [,ddl]'          ]  
      [ OTH          ]  
  
      INSERT=++C ,DELETE=++C ,REPLACE=++C  
      (-INS)      (-DEL)      (-REP)  
      (-INS)      (-DEL)      (-REP)  
      (xxxxx)      (xxxxx)      (xxxxx)
```

This is an informational message. It is shown only when COPYDIFF was specified.

When COPYDIFF is specified, COMPAREX writes to file SYSUT3 (assuming it can be opened successfully) any differing record from file SYSUT2. If any option to COPYDIFF is entered (PAN, LIB, GEM, IEBUPDTE, or OTH) and TEXT processing is in effect, differing records will be preceded by a formatted change control record.

The second line is displayed only if an option to COPYDIFF is specified. If COPYDIFF=PAN was specified, then INSERT, DELETE, and REPLACE are set to '+ +C'. If COPYDIFF=LIB was specified, then INSERT is set to '-INS', DELETE is set to '-DEL' and REPLACE is set to '-REP'. If COPYDIFF=GEM was specified, then INSERT is set to '- INS', DELETE is set to '- DEL' and REPLACE is set to '- REP'. If COPYDIFF=IEBUPDTE was specified, then INSERT is set to 'NUMBER', DELETE is set to 'DELETE', and REPLACE is set to 'CHANGE'. If COPYDIFF=OTH was specified, then the values entered by the user for INSERT, DELETE, and REPLACE are displayed here.

In addition to the formatted change control records such as (+ +C, -INS, etc.) the first record written is a directive to the library management system (PAN, LIB, GEM, IEBUPDTE, or OTH) to update the right module. For PAN, it is:

```
++UPDATE member,level  
++UPDATE member,level,TEMP
```

for LIB, it is:

```
-SEL member,pass,VERS=mddd  
-SEL member,pass,VERS=mddd,TEMP
```

- SEL member,VERS=mmdd
- SEL member,pass
- SEL member

for GEM, it is:

- SEL member
- SEL member,TEMP

for IEBUPDTE, it is:

./CPX CHANGE NAME=member,SEQFLD=(ddl,ddl)

or, if PRINT=FULL:

./CPX CHANGE NAME=member,SEQFLD=(ddl,ddl),LIST=ALL

and for OTH, it is:

?? member

If COPYDIFF=LIB is specified, suffixing records are written out. At the end of each updated member, it is:

-EMOD

and at the conclusion of all updates, it is:

-END

To review, examine the statements displayed to insure that the run is correct.

CPX16A

Message Format

CPX16A - SYSUT3 COPY FILE MISSING, INVALID, OR DUMMY - COPYDIFF NULLIFIED

This is an ACTION message. The user has entered the COPYDIFF keyword, but COMPAREX cannot write the SYSUT3 file. COMPAREX continues the run, without writing file SYSUT3.

ACTION: Examine the JCL and the user-entered keywords to determine the problem.

Possible reasons:

SYSUT3=DUMMY was specified. If the SYSUT3 file is desired, take out the SYSUT3=DUMMY specification.

COMPAREX could not open file SYSUT3. Examine the JCL. //SYSUT3 may have been missing or misspelled.

CPX16I

Message Format

CPX16I - SYSUT3=dsname.sysut3 DCB=(DSORG={PS},RECFM={F}[B][S],LRECL=n,BLKSIZE=n[,RKP=rkp,KEYLEN=n])
(IS) (V)
(DA) (U)

or

CPX16I - SYSUT3=dsname.sysut3 ACB=((ESDS),LRECL=n,CINV=n,[PASSWORD=xxx])[,RKP=rkp,KEYLEN=n])
(KSDS)
(RRDS)

This is an informational message. It is shown only when COPYDIFF is specified and the utility has successfully opened file SYSUT3.

The dsname.sysut3 is taken from the DSN parameter of the dataset allocated to SYSUT3 (//SYSUT3 DD DSNAME=...).

If the dataset organization is QSAM, or ISAM, the first option is shown. If the dataset organization is QSAM, DSORG=PS; if the dataset organization is ISAM, DSORG=IS; if the dataset organization is direct access, DSORG=DA.

RECFM, LRECL, BLKSIZE, RKP, and KEYLEN further describe the file.

If the dataset organization is VSAM, the ACB option is shown. ESDS, KSDS, RRDS, LRECL, CINV, RKP, and KEYLEN describe the VSAM file. If the RKP is shown with the ACB option and the RKP was extracted by COMPAREX from an operating system control block (such as a VSAM ACB), the relative key position value is relative to zero, even if the MODE=APPLICATIONS keyword has been entered.

To review: Examine the parameters of the message to determine if file SYSUT3 was correctly specified.

CPX18I

Message Format

CPX18I - SEGMENT SPECIFICATIONS IGNORED IN LIEU OF KEY SYNCHRONIZATION

This is an ACTION message. COMPAREX has found both one or more SEGMENT keywords and one or more KEY keyword specification. If SEGMENT synchronization is needed, KEY synchronization cannot also be used in the same COMPAREX job. If KEY synchronization is used, any SEGMENT keywords in the same job will be ignored. COMPAREX has processed this job using only the KEY keywords (and KEY1 and KEY2 pairs) for synchronization.

ACTION: Determine which type of synchronization is needed. If COMPAREX is to compare databases (or unloaded versions), SEGMENT synchronization may be needed; otherwise, KEY synchronization is needed. Respecify the synchronizing parameters to use only one type of synchronization.

CPX19I

Message Format

CPX19I - DATA BASE SEGMENTING:
SEGMENT=(d1,EQ,t'vvvv')
SEGMENT=(d1,EQ,t'vvvv',((A),d2,(((
(D)
(R)

This is an informational message. It is shown when one or more SEGMENT keyword specifications and no KEY keywords (or KEY1 and KEY2 pairs) are made. COMPAREX displays the parameters from the SEGMENT keywords, in the same order as specified.

To review: Examine the SEGMENT keywords and their parameters to insure that they have been correctly entered. If necessary, see "Chapter 6 - Data File Synchronization Keywords" on page 99 for more information about the SEGMENT keyword.

CPX20I

Message Format

CPX20I - CPXIFACE=CPXabcde(PANVALET::IDMS; 02/23/87-11:15-OS-6.1.0 -87/054)

This is an informational message. It is shown when COMPAREX has found the PAN, LIB, or OTH option to either SYSUT1 or SYSUT2 requesting that the COMPAREX interface be invoked to read from those files. The default load module name is CPXIFACE but any other one to eight character name may have been specified. The only requirement is that the load module name exist on an accessible library or anabend is certain.

The message reflects information about how the module was generated (and it doesn't have to be called CPXIFACE either) through a special call procedure. The standard calls of OPEN, SRCH, READ and CLOS are to read exotic files, but INFO is performed to extract date/time stamps and release level.

CPX21I

Message Format

CPX21I - SYSUT1=dsname.sysut1 DCB=(DSORG={PS},RECFM={F}[B][S],LRECL=n,BLKSIZE=n[,RKP=rkp,KEYLEN=n])
(IS) (V)
(DA) (U)
(PO)

or

CPX21I - SYSUT1=dsname.sysut1 ACB=((ESDS),LRECL=n,CINV=n[,PASSWORD=xxx] [,RKP=rkp,KEYLEN=n])
(KSDS)
(RRDS)

This is an informational message. It is shown when SYSUT1=DUMMY has not been specified, and after COMPAREX has successfully opened SYSUT1.

The dsname.sysut1 is taken from the dataset name of the file that is allocated to SYSUT1.

If the dataset organization is QSAM or ISAM, the first option is shown. The original DSORG found for the file is displayed. If the dataset organization is QSAM, DSORG=PS; if the dataset organization is ISAM, DSORG=IS; if the dataset organization is partitioned, DSORG=PO; if the dataset organization is direct access, DSORG=DA. RECFM, LRECL, BLKSIZE, RKP, and KEYLEN further describe the file.

If the dataset organization is VSAM, the ACB option is shown. ESDS, KSDS, RRDS, LRECL, CINV, RKP, and KEYLEN describe the VSAM file. If the RKP is shown with the ACB option and the RKP was extracted by COMPAREX from an operating system control block (such as a VSAM ACB), the relative key position value is relative to zero, even if the MODE=APPLICATIONS keyword has been entered.

To review: Examine the parameters of the message to determine if file SYSUT1 was correctly specified.

CPX22I

Message Format

CPX22I - SYSUT2=dsname.sysut2 DCB=(DSORG={PS},RECFM={F}[B][S],LRECL=n,BLKSIZE=n[,RKP=rkp,KEYLEN=n])
(IS) (V)
(DA) (U)
(PO)

or

CPX22I - SYSUT2=dsname.sysut2 ACB=((ESDS),LRECL=n,CINV=n[,PASSWORD=xxx] [,RKP=rkp,KEYLEN=n])
(KSDS)
(RRDS)

This is an informational message. It is shown when SYSUT2=DUMMY has not been specified, and after COMPAREX has successfully opened SYSUT2.

The dsname.sysut2 is taken from the dataset name of the file that is allocated to SYSUT2.

If the dataset organization is QSAM or ISAM, the first option is shown. The original DSORG found for the file is displayed. If the dataset organization is QSAM, DSORG=PS; if the dataset organization is ISAM, DSORG=IS; if the dataset organization is partitioned, DSORG=PO; if the dataset organization is direct access, DSORG=DA. RECFM, LRECL, BLKSIZE, RKP, and KEYLEN further describe the file.

If the dataset organization is VSAM, the ACB option is shown. ESDS, KSDS, RRDS, LRECL, CINV, RKP, and KEYLEN describe the VSAM file. If the RKP is shown with the ACB option and the RKP was extracted by COMPAREX from an operating system control block (such as a VSAM ACB), the relative key position value is relative to zero, even if the MODE=APPLICATIONS keyword has been entered.

To review: Examine the parameters of the message to determine if file SYSUT2 was correctly specified.

CPX23I

Message Format

CPX23I - SYNCHRONIZATION KEY TAKEN FROM SYSUT1 - KEY=(ddd, lll, C, A)

This is an informational message. It is shown only when COMPAREX finds no KEY or KEY1 specifications, and the SYSUT1 dataset organization, as specified in the CPX21I message, is ISAM or VSAM/KSDS. COMPAREX has generated a KEY based on the RKP and KEYLEN values specified in the CPX21I message, and the utility will use this KEY for key synchronization. in +2

To review: If key synchronization is desired, no change is necessary. If key synchronization is not desired for this execution of COMPAREX, the SYSUT1 keyword must be changed to specify a non-indexed file type. If the user has not specified MODE=SYSTEMS, the displacement shown with the KEY with message CPX23I will be relative to one (COMPAREX's default mode).

CPX24A

Message Format

CPX24A - TEXT SPECIFICATIONS IGNORED IN LIEU OF DATA - n

This is an ACTION message. The user has entered the TEXT keyword, but COMPAREX found a conflicting specification. COMPAREX has ignored the TEXT keyword, and the utility has processed this job using DATA comparison logic.

ACTION: Examine the reason code at the end of the message line, shown instead of n.

1. The user specified SYSUT2=DUMMY or file SYSUT2 could not be successfully opened. TEXT comparison logic requires both input files. Respecify DATA logic or specify a valid file for SYSUT2.
2. The user entered one or more KEY keywords (or KEY1 and KEY2 pairs). TEXT comparison logic does not use key synchronization. Delete the TEXT keyword or delete all KEY specifications.
3. The user entered one or more SEGMENT keywords. TEXT comparison logic does not use segment synchronization. Delete the TEXT keyword or delete all SEGMENT specifications.
4. The user entered at least one IDENTITY keyword or at least one DESEN (or DESEN1 or DESEN2). TEXT comparison logic uses neither IDENTITYs nor DESENs. Delete the TEXT keyword or eliminate IDENTITYs and/or DESENs.
5. The user entered more than one FIELD keyword or COMPAREX compiled more than one FIELD from the user-specified FIELDS and MASKs. TEXT comparison logic uses only one FIELD. If any option to TEXT is specified (such as TEXT=COBOL) except for TEXT=REPORT, COMPAREX generates a FIELD specification, and the user may not enter any FIELD keyword. With the TEXT=REPORT option, COMPAREX does not generate any FIELD and the user may not enter any FIELD keyword. Delete the TEXT keyword or delete all but one FIELD specification.

CPX25I

Message Format

CPX25I - DIRECTORY[={SPF}]
 [{PDF}]
 [{USER}]

or

TEXT [=xxxx] ,MLC=n,BUFF=nn,FRAME={YES}
[SQUEEZE=t'vv'] {NO}
 {NUM}

or

DATA [=CSECT] [,BUFF=nn,]FORMAT=xy,INTERLEAVE=n

(FORMAT EXPLANATION: {FULL SYSUT1} {FOLLOWED BY} {FULL SYSUT2})
 {DIFFERING LINES OF SYSUT1} {INTERLEAVED WITH} {DIFFERING LINES OF SYSUT2}

This is an informational message. It is always issued. If DIRECTORY is specified and not nullified because SYSUT1 does not point to a directory-embedded dataset, then the first option is shown.

If TEXT is specified and not nullified (see message CPX24A), then the second option is shown. If SQUEEZING is specified or implied by option to TEXT, then all SQUEEZE specifications are shown one per line.

In all other cases, the third option is shown.

To review: Examine the parameters displayed to insure that the keywords were interpreted as desired.

CPX26I

Message Format

CPX26I - SKIP RECORD PROCESSING: SYSUT1(n1)/SYSUT2(n2)

This is an informational message. COMPAREX has found either a SKIPUT1 or SKIPUT2 specification, and the utility has opened and skipped over (read without passing any record to the compare routines) the required number of records. If directory-embedded datasets are being read, this message is issued at the beginning of each member, after the number of records specified by message CPX09I has been bypassed for that member.

If the number of records on the file is less than the number of records to be skipped over, n1 and n2 show the number of records on the file; otherwise, n1 and n2 show the number of records skipped over.

To review: Compare the numbers of records skipped over to the numbers given in message CPX09I. If the numbers are different, one or both input files are shorter than the user anticipated, and the user may determine that the wrong files were specified.

CPX27I

Message Format

CPX27I - PRINTING OF SYSUT1 ONLY INVOKED

If SYSUT2 should be present, this is an ACTION message.
If SYSUT2=DUMMY has been specified, or COMPAREX is unable to open SYSUT2 and HALT=NO has been specified, or after successfully opening file SYSUT2, finds the file to be a dummy file (*the JCL specifies //SYSUT2 DD DUMMY or //SYSUT2 DD DSN=NULLFILE*), the utility issues this message.

When this message is issued, COMPAREX sends no records to the comparison routines. Instead, any SKIPUT1 keyword is used to skip over records, and any filtering keywords select or reject records. Selected records from SYSUT1 are printed on the Differences Report.

ACTION: Point to correct SYSUT2 file.

If SYSUT2 should not be present, this is an informational message.

To review: No error correction is necessary.

CPX30A

Message Format

CPX30A - EXECUTION HALTED BY REQUEST - FUNCTION TERMINATED - RETURN CODE = 16

This is an ACTION message. It is directly related to the setting of keyword HALT. Either HALT=YES or HALT=COND with syntax errors has been specified. No files will be read and compared.

ACTION: If HALT=YES was intentionally specified, you now know what happens. If HALT=COND was specified (*could be part of installation defaults*) then the syntax errors underscored and denoted by ERROR? should be corrected for the subsequent rerun.

CPX31A

Message Format

CPX31A - FILE ORGANIZATIONS NOT COMPATIBLE - n - FUNCTION TERMINATED - RETURN CODE = 16

This is an ACTION message. An attempt was made to compare two files that have no chance of being processed together by COMPAREX.

ACTION: Examine the reason code shown instead of n.

1. SYSUT1 is specified as DUMMY and SYSUT2 points to a directory-embedded dataset. Either reverse the situation or let them both point to directory-embedded datasets.
2. SYSUT1 points to a sequential dataset and SYSUT2 points to a directory-embedded dataset. Let both point to sequential files or both point to directory-embedded datasets.
3. SYSUT1 and SYSUT2 both point to directory-embedded datasets but both are PAN or both are LIB and DIRECTORY (such as DIR=PDF) has not been specified.
4. SYSUT1 points to a directory-embedded dataset but SYSUT2 is sequential. Either set SYSUT2 to DUMMY or also let it point to a compatible directory-embedded dataset.

Messages During Processing and End of Job

Messages issued by COMPAREX while it is reading records and comparing them are used to show the detailed results of the comparison process. They are, with the exception of messages CPX35A and CPX37A, informational messages, used to give the user information about the records displayed on the Differences Report. Messages CPX35A and CPX37A are issued prior to an abnormal job termination; the user must take an action.

CPX35A

Message Format

CPX35A - KEY SYNCHRONIZATION VALUES TOO LARGE FOR FILE - FUNCTION TERMINATED - RETURN CODE = 16

This is an ACTION message. The user has entered one or more KEY keywords (or KEY1 and KEY2 pairs), and the displacement given on at least one KEY was greater than the length of the record being processed. This means that the starting position (displacement) of the KEY was beyond the end of the record being read when the message was issued. COMPAREX terminates, showing a return code of 16.

ACTION: Change the displacement on the KEY keyword so that the value is less than the length of the shortest record on the file. If the short records can be identified with a logical test, the user could FILTER them out (only records that are FILTER'ed in are synchronized by KEY).

CPX36A

Message Format

CPX36A - {KEY} OUT OF SPECIFIED SEQUENCE - RECORD nn1 [(RBA=nn2)] ON FILE SYSUT(1)
(SEGMENT) (2)

This is an ACTION message. It will only appear once per execution at most. At least one KEY (or KEY1/KEY2 pair) or SEGMENT has been specified which contained an Ascending or Descending control field and it was found not to be in that sequence. The message is issued and then the offending record is displayed right after it.

ACTION: Either correct the specification for KEY or SEGMENT, sort the file into a usable order, or try Random as opposed to Ascending or Descending order.

CPX37A

Message Format

CPX37A - DB SEGMENTING SYNCH. VALUES TOO LARGE FOR FILE - FUNCTION TERMINATED - RETURN CODE = 16

This is an ACTION message. The user has entered one or more SEGMENT keywords, and the displacement for a synchronizing key on at least one SEGMENT was greater than the length of the record being processed. This means that the starting position (displacement) of the SEGMENT was beyond the end of the record being read when the message was issued. COMPAREX terminates, showing a return code of 16.

ACTION: Change the displacement on the SEGMENT keyword so that the value is less than the length of the shortest record on the file. If the short records can be identified with a logical test, the user could FILTER them out (only records that are FILTER'ed in are synchronized by SEGMENT).

CPX39A

Message Format

CPX39A - {TEXT} RECORDS TOO LARGE FOR BUFFER - (xxxxxxx,nnn) - FUNCTION TERMINATED - RETURN CODE = 16
(DATA)

When parsing CSECT's, the BUFF parameter defaults to 256. It is also explicitly recommended that BUFF be set higher such as:

BUFF=1000

If the BUFFER cannot fully contain both load modules (*at buffer stocking time*) then the comparison is terminated with this message.

The xxxxxxx part of the message can be either a CSECT name or

\$RECORD\$

If it is a CSECT name, then the ESD entry for this CSECT stated that it would spill beyond the Buffer capacity. The value for nnn is the high decimal number of the beginning address plus the CSECT size that would otherwise overflow (*SOC4*) the Buffer.

If it is \$RECORD\$, the message is displayed when actually processing the undefined length record. Again, the nnn is the high decimal number of the beginning address plus the record length that would otherwise overflow the Buffer.

This is an ACTION message. The user has attempted to compare files with very large records under TEXT logic, DATA with Random keys, or Csect Parsing, and not adequately increased the working buffer (BUFF).

ACTION: Either eliminate Random keys or specify a larger value for BUFF. The BUFF keyword can go to 1024 (*actually 16 Meg with MODE=SYSTEMS*).

CPX40A

Message Format

CPX40A - NO EXTERNAL SYMBOL DICTIONARY ENTRIES - FUNCTION TERMINATED - RETURN CODE = 16

This is an ACTION message. The user has attempted to compare load modules via DATA=CSECT (csect parsing) and COMPAREX determined that one of the files is not a load module. COMPAREX checks that the first record of the file is a legal External Symbol Dictionary (ESD) entry. If the TEST option has been turned on at compile and linkage edit time, then the first record(s) is not an ESD and this message is issued just before termination with return code 16.

ACTION: Either point to load module libraries, use regular DATA logic (not DATA=CSECT), or filter out the load module that has the TEST option turned on.

CPX41I

Message Format

CPX41I - CSECT=csectnam DIF ONE

This is an informational message. DATA=CSECT (*csect parsing*) has been specified. COMPAREX has determined that csectnam exists in the load module that SYSUT1 points to but does not exist in SYSUT2.

To review: No error correction is necessary. Use this information to check the effectiveness of the comparison.

CPX42I

Message Format

CPX42I - CSECT=csectnam DIF T W O

This is an informational message. DATA=CSECT (*csect parsing*) has been specified. COMPAREX has determined that csectnam exists in the load module that SYSUT2 points to but does not exist in SYSUT1.

To review: No error correction is necessary. Use this information to check the effectiveness of the comparison.

CPX51I

Message Format

CPX51I - RECORD NUMBER nn1 [(RBA=nn2)] ON FILE SYSUT1

or

CPX51I - CSECT=csectnam

This is an informational message. The record that immediately follows this message on the Differences Report was read from file SYSUT1 and it was selected to be printed on the Differences Report based on the values of the PRINT and FORMAT keywords. The logical record number is shown instead of 'nn1.'

If the dataset organization for file SYSUT1 is VSAM, the relative byte address of the first byte of the record is shown instead of 'nn2' and the 'RBA=' literal is shown.

If DATA=CSECT (csect parsing) has been specified, the second form of the message is used. The specified csect 'csectnam' has been isolated in each file and differences have been found. The csect from SYSUT1 will be displayed.

To review: No error correction is necessary. Use this record to check the effectiveness of a program or a modification.

CPX52I

Message Format

CPX52I - RECORD NUMBER nn1 [(RBA=nn2)] ON FILE SYSUT2 [IDENTITY=i,][FIELD=f]

or

CPX52I - CSECT=csectnam

This is an informational message. The record that this message refers to was read from file SYSUT2 and selected to be printed on the Differences Report based on the values of the PRINT and FORMAT keywords. The logical record number is shown instead of 'nn1.'

This SYSUT2 record has been paired with a SYSUT1 record for comparison, synchronized by KEYS, SEGMENTS, or by physical locations on the files.

If the dataset organization for file SYSUT2 is VSAM, the relative byte address of the first byte of the record is shown instead of 'nn2' and the 'RBA=' literal is shown.

If FIELDS or IDENTITYs were specified and message CPX12I was issued, the sequence number from message CPX12I is shown for the IDENTITY and FIELD identified with the record. i shows the IDENTITY sequence number, and f shows the sequence number of the FIELD where the first inequality was found during the comparison to the SYSUT1 record.

If FLDSONLY has been specified or if COMPAREX has generated a FLDSONLY specification (see message CPX11I), those bytes that differ in the selected FIELDS are underscored with the DASH character.

If FLDSONLY has not been specified or generated, all differing bytes are underscored with the DASH character.

If NIBBLE has been specified, COMPAREX underscores only differing half-bytes with the DASH character.

If the SYSUT2 record is longer than the paired SYSUT1 record, COMPAREX underscores the excess bytes on the SYSUT2 record with the PLUS character.

If DATA=CSECT (csect parsing) has been specified, the second form of the message is used. The specified csect 'csectnam' has been isolated in each file and differences have been found. The csect from SYSUT2 will be displayed and the differences underscored.

To review: No error correction is necessary. Use this record to check the effectiveness of a program or a modification.

CPX56I

Message Format

CPX56I - EXTRA RECORD NUMBER nn1 [(RBA=nn2)] ON FILE SYSUT1

This is an informational message. The record that immediately follows this message on the Differences Report was read from file SYSUT1. File SYSUT2 has come to end of file and file SYSUT1 has not come to end of file. This message presents a record that is on the end of file SYSUT1 that is not able to be synchronized to a record from file SYSUT2.

The logical record number is shown instead of 'nn1.'

If the dataset organization for file SYSUT1 is VSAM, the relative byte address of the first byte of the record is shown instead of 'nn2' and the 'RBA=' literal is shown.

To review: No error correction is necessary. Use this record to check the effectiveness of a program or a modification.

CPX57I

Message Format

CPX57I - EXTRA RECORD NUMBER nn1 [(RBA=nn2)] ON FILE SYSUT2

This is an informational message. The record that immediately follows this message on the Differences Report was read from file SYSUT2. File SYSUT1 has come to end of file and file SYSUT2 has not come to end of file. This message presents a record that is on the end of file SYSUT2 that is not able to be synchronized to a record from file SYSUT1.

The logical record number is shown instead of 'nn1.'

If the dataset organization for file SYSUT2 is VSAM, the relative byte address of the first byte of the record is shown instead of 'nn2' and the 'RBA=' literal is shown.

To review: No error correction is necessary. Use this record to check the effectiveness of a program or a modification.

CPX61I

Message Format

CPX61I - KEY SYNCHRONIZATION MISMATCH - RECORD nn1 [(RBA=nn2)] ON FILE SYSUT1

This is an informational message. The record that immediately follows this message on the Differences Report was read from file SYSUT1. File synchronization is being done, for this execution of COMPAREX, by KEYs, and this following SYSUT1 record is not matched by KEY to any SYSUT2 record.

The logical record number is shown instead of 'nn1.'

If the dataset organization for file SYSUT1 is VSAM, the relative byte address of the first byte of the record is shown instead of 'nn2' and the 'RBA=' literal is shown.

To review: No error correction is necessary. Use this record to check the effectiveness of a program or a modification.

CPX62I

Message Format

CPX62I - KEY SYNCHRONIZATION MISMATCH - RECORD nn1 [(RBA=nn2)] ON FILE SYSUT2

This is an informational message. The record that immediately follows this message on the Differences Report was read from file SYSUT2. File synchronization is being done, for this execution of COMPAREX, by KEYs, and this following SYSUT2 record is not matched by KEY to any SYSUT1 record.

The logical record number is shown instead of 'nn1.'

If the dataset organization for file SYSUT2 is VSAM, the relative byte address of the first byte of the record is shown instead of 'nn2' and the 'RBA=' literal is shown.

To review: No error correction is necessary. Use this record to check the effectiveness of a program or a modification.

CPX64I

Message Format

CPX64I - SEGMENT SYNCHRONIZATION MISMATCH - RECORD nn1 [(RBA=nn2)] ON FILE SYSUT1

This is an informational message. The record that immediately follows this message on the Differences Report was read from file SYSUT1. File synchronization is being done, for this execution of COMPAREX, by SEGMENTS, and this following SYSUT1 record is not matched by SEGMENT to any SYSUT2 record.

The logical record number is shown instead of 'nn1.'

If the dataset organization for file SYSUT1 is VSAM, the relative byte address of the first byte of the record is shown instead of 'nn2' and the 'RBA=' literal is shown.

To review: No error correction is necessary. Use this record to check the effectiveness of a program or a modification.

CPX65I

Message Format

CPX65I - SEGMENT SYNCHRONIZATION MISMATCH - RECORD nn1 [(RBA=nn2)] ON FILE SYSUT2

This is an informational message. The record that immediately follows this message on the Differences Report was read from file SYSUT2. File synchronization is being done,

for this execution of COMPAREX, by SEGMENTS, and this following SYSUT2 record is not matched by SEGMENT to any SYSUT1 record. The logical record number is shown instead of 'nn1.' If the dataset organization for file SYSUT2 is VSAM, the relative byte address of the first byte of the record is shown instead of 'nn2' and the 'RBA=' literal is shown.

To review: No error correction is necessary. Use this record to check the effectiveness of a program or a modification.

CPX66A

Message Format

CPX66A - SEGMENT ID NOT MATCHABLE - RECORD nnn ON FILE SYSUT(1)
(2)

This is an ACTION message. The user has specified several SEGMENT keywords and each one describes a particular Segment identification. A record is read from either SYSUT1 or SYSUT2 which does not match any of the SEGMENT specifications, then this message is issued followed by a full display of the offending record. Then the record is discarded from further processing as if it satisfied a FILTOROUT. No statistics are gathered for this event. The purpose of this is to allow more accurate synchronization. Unanticipated or misspelled segment types can have unpredictable results. The logical record number is shown instead of 'nnn.'

To review: Determine if the unanticipated record really is a segment type that is expected. Specify additional SEGMENT keywords to cover this situation.

CPX67I

Message Format

CPX67I - MAXDIFF INVOKED[, CONTINUING WITHOUT PRINTING BY REQUEST]

This is an informational message. MAXDIFF has been specified, and the number of differences specified by the MAXDIFF keyword's parameter has been printed on the Differences Report. After message CPX67I is printed, no further records from file SYSUT1 or file SYSUT2 will be shown on the Differences Report.

If CONTINUE has also been specified, the second line of message CPX67I is shown. This indicates that COMPAREX will continue to read the input files and compare them without writing any records on the Differences Report. If CONTINUE is specified, the end-of-processing totals will reflect the total number of records on the input files and the total number of differences found.

If CONTINUE has not been specified, COMPAREX will stop processing immediately after message CPX67I is printed. This means that COMPAREX will close its input files and issue its end-of-processing messages immediately.

To review: Examine the records on the Differences Report. If more records are needed to check the effectiveness of a program or a modification, change the MAXDIFF parameter to a higher number for subsequent executions.

CPX69I

Message Format

CPX69I - STOPAFT INVOKED

This is an informational message. Either STOPAFT has been specified and the number of records specified by the STOPAFT keyword's parameter has been reached, or COMPAREX has read 999,999,999,999 records from one of the input files.

COMPAREX, after it writes message CPX69I, closes its input files and issues its end-of-processing messages. In addition, COMPAREX terminates, showing a return code of 8.

To review: Examine the records on the Differences Report. If COMPAREX needs to read more records to check the effectiveness of a program or a modification, change the STOPAFT parameter to a higher number for subsequent executions.

CPX71I

Message Format

```
CPX71I - END OF {DATA}      ON FILE SYSUT1
              {TEXT}
              {DIRECTORY}
```

This is an informational message. COMPAREX has detected an end-of-file indication on file SYSUT1. No records on the Differences Report that follow this message are from file SYSUT1.

If the input files are not directory-embedded and message CPX25I specified TEXT comparison logic, message CPX71I shows the 'TEXT' literal; if the input files are not directory-embedded and message CPX25I specified DATA comparison logic, message CPX71I shows the 'DATA' literal.

If the input files are directory-embedded datasets, this message can appear more than once during the run. When COMPAREX detects the end of a member, the utility issues message CPX71I, showing the 'DATA' or 'TEXT' literal (to indicate the type of comparison logic specified with message CPX25I). When COMPAREX detects the end of all members, the utility issues message CPX71I, showing the 'DIRECTORY' literal.

To review: No error correction is necessary. Use this message and its location on the Differences Report to evaluate the correctness of the COMPAREX run. .

CPX72I

Message Format

```
CPX72I - END OF {DATA}      ON FILE SYSUT2
              {TEXT}
              {DIRECTORY}
```

This is an informational message. COMPAREX has detected an end-of-file indication on file SYSUT2. No records on the Differences Report that follow this message are from file SYSUT2.

If the input files are not directory-embedded and message CPX25I specified TEXT comparison logic, message CPX72I shows the 'TEXT' literal; if the input files are not directory-embedded and message CPX25I specified DATA comparison logic, message CPX72I shows the 'DATA' literal.

If the input files are directory-embedded datasets, this message can appear more than once during the run. When COMPAREX detects the end of a member, the utility issues message CPX72I, showing the 'DATA' or 'TEXT' literal (to indicate the type of comparison logic specified with message CPX25I). When COMPAREX detects the end of all members, the utility issues message CPX72I, showing the 'DIRECTORY' literal.

To review: No error correction is necessary. Use this message and its location on the Differences Report to evaluate the correctness of the COMPAREX run.

CPX74I

Message Format

CPX74I - BYTES UNDERScoreD (u1[,u2])

This is an informational message. It is not shown if message CPX25I specified TEXT processing.

COMPAREX tallies the number of bytes underscored with the DASH character on the Differences Report, and the utility shows this number as u1.

If any SYSUT2 record on the Differences Report was longer than the SYSUT1 record to which it was paired, COMPAREX adds the number of excess bytes on the SYSUT2 record to counter u2. If, at the end of the COMPAREX run, counter u2 is not zero, COMPAREX shows counter u2 as the second number of message CPX74I.

To review: No error correction is necessary. Use these counters to evaluate the effectiveness of a program or a modification.

CPX75I

Message Format

CPX75I - RECORDS PROCESSED: SYSUT1(n1)/SYSUT2(n2)[/SYSUT3(n3)], DIFFERENCES(d0[,d1,d2])

EXPLANATION - d0 RECORDS DIFFER THAT SYNCHRONIZED TOGETHER
d1 RECORD WAS CONSIDERED INSERTED ON SYSUT1
d2 RECORDS WERE CONSIDERED INSERTED ON SYSUT2

PASS	FAIL	STATISTICS
f1	f2	FILTERIN=(. . .
i1	i2	IDENTITY=(. . .

This is an informational message. It appears at the end of every COMPAREX run where DIRECTORY is not specified.

COMPAREX tallies the number of records read from file SYSUT1, and the utility shows that number as n1. This counter includes any records that were skipped over as a result of a SKIPUT1 keyword.

COMPAREX tallies the number of records read from file SYSUT2, and the utility shows that number as n2. This counter includes any records that were skipped over as a result of a SKIPUT2 keyword.

If COPYDIFF was specified and COMPAREX was able to successfully open file SYSUT3, COMPAREX shows the number of records written to file SYSUT3 as n3.

If TEXT processing was specified by message CPX25I, d0, d1, and d2 are calculated in this way:

d0 contains the number of records in each block that differ on a one-to-one basis. For example, if a block of records from file SYSUT1 is identified as differing from a block of records from file SYSUT2 and one block contains four records and one block contains five records, the d0 counter is increased by four. The extra record (the fifth record that did not differ on a one-to-one basis) is used to increase either counter d1 or counter d2 in this way:

If the block that contained more records came from file SYSUT1, the counter d1 is increased by the number of records in the block from file SYSUT1 minus the number of records in the block from file SYSUT2 (in our example, 5 minus 4 equals 1; counter d1 is increased by 1).

If the block that contained more records came from file SYSUT2, the counter d2 is increased by the number of records in the block from file SYSUT2 minus the number of records in the block from file SYSUT1.

If message CPX25I specified DIRECTORY processing and COMPAREX determined that both input files were directory-embedded, this message does not appear. Instead, COMPAREX issues message CPX78I.

If both input files are directory-embedded and message CPX25I did not specify DIRECTORY processing, message CPX75I appears at the end of each member, showing

the number of input records, the number of any SYSUT3 records written, and the number of differences found for that member. Then, at the end of the processing, COMPAREX issues message CPX78I to show the number of partitioned dataset members processed during the run.

If KEY or SEGMENT synchronization was specified for this run, d0 contains the number of pairs of records which were synchronized by KEY or SEGMENT and some difference was detected, d1 contains the number of records from file SYSUT1 that were not synchronized to any SYSUT2 record (records inserted into SYSUT1 or on the end of SYSUT1), and d2 contains the number of records from file SYSUT2 that were not synchronized to any SYSUT1 record (records inserted into SYSUT2 or on the end of SYSUT2).

If KEY or SEGMENT synchronization was not specified for this run, d0 contains the number of pairs of records which were synchronized by having the same physical location on the files and some difference was detected, d1 contains the number of records on the end of file SYSUT1 that could not be paired to SYSUT2 records (extra records on SYSUT1), and d2 contains the number of records on the end of file SYSUT2 that could not be paired to SYSUT1 records (extra records on SYSUT2).

To review: No error correction is necessary. Use these counters to evaluate the effectiveness of the COMPAREX run.

CPX76I

Message Format

CPX76I - UNUSABLE FIELD COMPARISONS(n1)/UNUSABLE IDENTITIES(n2)/UNUSABLE SEGMENTS(n3)/UNUSABLE DESENSITIZERS(n4)

This is an informational message. It is shown only if COMPAREX found one or more FIELD, IDENTITY, DESEN, or SEGMENT specifications, and the displacement given on any of these keywords went beyond the length of one or more input records.

If COMPAREX found one or more FIELD (or FIELD1 and FIELD2 pair) specifications, and COMPAREX determined that the displacement of the FIELD was beyond the end of the input record, COMPAREX adds 1 to a counter for each FIELD on each record where this out-of-bounds situation occurs. COMPAREX shows this counter as n1.

If COMPAREX found one or more IDENTITY specifications, and COMPAREX determined that the displacement of the IDENTITY was beyond the end of the input record, COMPAREX adds 1 to a counter for each IDENTITY on each record where this out-of-bounds situation occurs. COMPAREX shows this counter as n2.

If COMPAREX found one or more SEGMENT specifications, and COMPAREX determined that the displacement of the SEGMENT ID was beyond the end of the input record, COMPAREX adds 1 to a counter for each SEGMENT where this out-of-bounds situation occurs. COMPAREX shows this counter as n3. Note that if COMPAREX finds a SEGMENT where the starting position of any synchronization key (displacement) is out-of-bounds, COMPAREX immediately stops processing, issuing message CPX37A; if message CPX76I has been issued, COMPAREX did not find a SEGMENT where the displacement was out-of-bounds.

If COMPAREX found one or more DESEN (or DESEN1 or DESEN2) specifications, and COMPAREX determined that the desensitizing field would extend beyond the end of the input record, COMPAREX adds 1 to a counter for each DESEN where this out-of-bounds situation occurs. COMPAREX shows this counter as n4.

To review: No error correction is necessary. Use these counters to evaluate the correctness of the COMPAREX run.

CPX77I

Message Format

CPX77I - RECORDS REJECTED BY FILTERS: SYSUT1(n1)/SYSUT2(n2) - UNUSABLE FILTERS(n3)

This is an informational message. If COMPAREX found one or more filtering specifications, this message is issued.

If a record from file SYSUT1 was read and it was not passed to the comparison routines because of the filtering tests, COMPAREX adds 1 to counter n1.

If a record from file SYSUT2 was read and it was not passed to the comparison routines because of the filtering tests, COMPAREX adds 1 to counter n2.

If COMPAREX determined that the displacement of a FILTER was beyond the end of the input record, COMPAREX adds 1 to a counter for each FILTER on each record where this out-of-bounds situation occurs. COMPAREX shows this counter as n3.

If the MEMBER, or M, option is used in a filtering test, that FILTER does not cause any increases to the counters shown with message CPX77I. Instead, the counters shown with message CPX78I are affected by FILTERs with the MEMBER option.

To review: No error correction is necessary. Use these counters to evaluate the correctness of the COMPAREX run.

CPX78I

Message Format

CPX78I - MEMBERS PROCESSED: SYSUT1(n1)/SYSUT2(n2), DIFFERENCES(d0[,d1,d 2])
[- REJECTED BY FILTERS: SYSUT1(f1)/SYSUT2(f2)]

EXPLANATION - d0 MEMBERS DIFFER THAT SYNCHRONIZED TOGETHER
d1 MEMBER WAS CONSIDERED INSERTED ON SYSUT1
d2 MEMBERS WERE CONSIDERED INSERTED ON SYSUT2

This is an informational message. It is issued when both file SYSUT1 and file SYSUT2 are directory-embedded datasets.

The number of members in SYSUT1 is n1 and the number of members in SYSUT2 is n2.

Counter d0 reflects the number of matched member names where there was at least one difference. Counter d1 is the number of members on SYSUT1 that had no corresponding member name in SYSUT2. Counter d2 is the number of members on SYSUT2 that had no corresponding member name in SYSUT1.

If one or more FILTER keywords contained the MEMBER option, COMPAREX tallies the number of members FILTER'ed out as a result of these FILTERs. COMPAREX shows the number of members FILTER'ed out from the SYSUT1 file as f1, and COMPAREX shows the number of members FILTER'ed out from the SYSUT2 file as f2.

To review: No error correction is necessary. Use these counters to evaluate the effectiveness of the COMPAREX run.

CPX80I

Message Format

CPX80I - TIME OF DAY AT END OF JOB: hh:mm:ss CONDITION CODE ON EXIT: c

This is an informational message. It is issued at the end of every COMPAREX run. The processor's clock is accessed, just before the message is written onto SYSPRINT, and the time is shown. Hours are between 0 and 23; hours between 0 and 11 are morning (a.m.) hours, and hours between 12 and 23 are afternoon and evening (p.m.) hours (to find the 12-hour clock time for p.m. hours, subtract 12 from the hours figure if the hours figure is 13 or greater).

The condition code or return code is also shown. The values for c are:

0 - a comparison was performed but no differences were found. If file SYSUT1 contained no records and file SYSUT2 was not opened for any reason, this zero condition code is also used.

If differences in the two files exist but the user has FILTER'ed out all differing records or MASK'ed all differing fields or specified one or more FIELD statements and the differing data did not occur in any of these FIELDS, this zero condition is used. No records are shown on the Differences Report with the zero condition code.

4 - a comparison was performed and at least one difference was found. If file SYSUT1 contained one or more records and file SYSUT2 was not opened for any reason, this four condition code is also used.

8 - a comparison was performed, or file SYSUT1 contained one or more records and file SYSUT2 was not opened for any reason, but either

(a) The STOPAFT number, as shown with message CPX04I, was reached on one input file (that number of records was read after the SKIPUT number was bypassed).

(b) The MAXDIFF number, as shown with message CPX04I, was reached (that number of differences was printed on the Differences Report) but CONTINUE was not shown with message CPX04I.

16 - a serious error occurred during the comparison process. This error caused COMPAREX to stop processing and to print another message. Look for message CPX30A, CPX31A, CPX35A, CPX37A, CPX39A, CPX40A, CPX90A, CPX91A, CPX92A, CPX93A, CPX94A, or CPX99A. Then, refer to the information in this chapter about that message to identify the serious error and to resolve it.

CPX90A

Message Format

CPX90A - UNABLE TO OPEN FILE (SYSUT1 - FUNCTION TERMINATED - RETURN CODE = 16)
(SYSUT2 - ASSUME PRINT MODE)

This is an ACTION message. COMPAREX has tried to open a specified file, but the open has not been successful.

If COMPAREX has not been able to open file SYSUT1, the utility terminates, showing a return code of 16.

If COMPAREX has not been able to open file SYSUT2, and the user specified HALT=NO, the utility continues as a print utility, printing SYSUT1 records onto the Differences Report.

In the case of the COMPAREX interface (specifying PAN, LIB, or OTH within the SYSUT1 or SYUST2 keywords), and the particular interface module (in the form CPXabcde) has not been generated properly or there is an open error of some sort, the message is preceded by a 'feedback' message in the form:

CPXabcde/plo - NOT GENERATED PROPERLY
or
CPXabcde/feedback error message unique to interface

which gives assisting diagnostic information as to what happened at this point in OPEN'ing.

ACTION: Determine why COMPAREX was unable to open the file or inspect CPXIFACE to see if requested interface was generated properly.

CPX91A

Message Format

CPX91A - VSAM LOGICAL ERROR EXIT - SYSUT(1),RC=rc,RBA=n1,RECNO=n2 - FUNCTION TERMINATED - RETURN CODE = 16
(2)
(3)

This is an ACTION message. COMPAREX has tried to read or write a VSAM file, and the VSAM access routines have returned information that the read or the write was not completed.

ACTION: Refer to the VSAM Programmer's Guide, using the RC, RBA, and RECNO data.

CPX92A

Message Format

CPX92A - SYNCHRONOUS ERROR EXIT - SYSUT{1},RECNO=nnn - FUNCTION TERMINATED - RETURN CODE = 16
{2}
{3}

This is an ACTION message. COMPAREX has discovered a physical I/O error. The file name is shown, and nnn gives the physical record number on the file where the I/O error occurred. COMPAREX terminates, showing a return code of 16.

ACTION: If the physical I/O error occurred on file SYSUT1 or file SYSUT2 (COMPAREX's input files), use IBM utilities to copy the data to another file. If the physical I/O error occurred on file SYSUT3 (COMPAREX's output file), rerun COMPAREX.

CPX93A

Message Format

CPX93A - COMPAREX INTERFACE NOT PRESENT - FUNCTION TERMINATED - RETURN CODE = 16

This is an ACTION message. An attempt was made to access the load module, as specified by message CPX20I, and it was unsuccessful.

ACTION: Generate a COMPAREX interface module onto an accessible library or point to a previously generated module via the CPXIFACE keyword. Rerun COMPAREX.

CPX94A

Message Format

CPX94A - INTERFACE ERROR EXIT SYSUT(1), (SRCH), RC=n, RECNO=n - FUNCTION TERMINATED - RETURN CODE = 16
(2) (READ)

This is an ACTION message. The COMPAREX interface returned an unexpected condition code. The most common error is requesting to read a member that does not exist on the library, or, if under LIBRARIAN, not archived at the requested relative level. Other incidental causes have to do with reading a module from PANVALET, GEM, or LIBRARIAN, and having an INCLUDE'd member be missing from the library or be at a different status.

In almost all cases, the message is preceded by a 'feedback' message from the particular interface module (usually CPXIFACE). It is in the form:

CPXIFACE/feedback error message unique to interface

which gives assisting diagnostic information as to what happened at this point in SRCH'ing or READ'ing.

ACTION: If the return code (RC) is 12, 16, or 20, a very serious error was encountered. If the return code is 8, an I/O error has occurred. If the return code is 4, a particular member was not found. Correct and rerun.

CPX97A

Message Format

CPX97A - PROGRAM INTERRUPT ENCOUNTERED - FUNCTION TERMINATED - RETURN CODE = 16

This is an ACTION message. An abnormal end (abend) has occurred within COMPAREX or the (CPXIFACE) interface. SPIE (*Specify Program Interrupt Exit*) has been turned on early in COMPAREX and it intercepted a potential S0C1 through S0CF abend. Not all abend situations are caught however. S80A - insufficient storage for BUFF getmain - will not be intercepted. The dump has been suppressed but if the user wants to get a dump (//SYSUDUMP DD SYSOUT=*) on a rerun, then specify:

KILLSPIE=YES

The SPIE will not be issued, and the dump will be generated. The KILLSPIE keyword is intentionally not documented in this manual. Very likely, the reason the user read this is because this message came out.

Abends can happen under these known conditions:

1. specifying Zoned or Packed on a FIELD, FIELD1, or FIELD2 and it is not Zoned or Packed.
2. specifying Zoned or Packed on a KEY, KEY1, or KEY2 and it is not Zoned or Packed.

3. specifying COPYDIFF=IEBUPDTE when comparing PDS's and they do not have their sequence numbers in the columns as specified in the SEQFLD subkeyword to COPYDIFF.

ACTION: Either correct the FIELD, KEY, or SEQFLD specifications and rerun, or specify KILLSPIE on the rerun and send the dump to your marketing representative.

CPX99A

Message Format

CPX99A - LICENSE EXPIRED - CONTACT your marketing rep.!!!

This is an ACTION message. A user tried to execute COMPAREX, but the utility's license had expired. Return code 16 is set at end of job.

ACTION: Contact your marketing representative as described in the above message. If that does not get satisfactory results, contact:

SERENA Consulting
Post Office Box 117039
Burlingame, California 94010 USA
Telephone (415) 347-0100 Collect!

Alternate Message Format

CPX99A - INSUFFICIENT VIRTUAL STORAGE - FUNCTION TERMINATED

This is an ACTION message. A user tried to execute COMPAREX, but there is insufficient virtual storage.

ACTION: Either increase the size of your region or lower your specification for BUFF.

Chapter 11 - Examples

The following examples illustrate the coding of keywords to meet various testing situations.

Select One Account - FILTERIN

We want to select a special test file from the master file containing only account 34567-8:

```
//COMPARE PROC
//COMPAREX EXEC PGM=COMPAREX,REGION=256K
//SYSPRINT DD SYSOUT=*
//          PEND
//*
//EXTRACT EXEC COMPARE
//SYSUT1    DD DUMMY
//SYSUT2    DD DISP=SHR,DSN=somnode.MASTER.FILE
//SYSUT3    DD DISP=(,CATLG,DELETE),DSN=somnode.EXTRACT.FILE,
//          UNIT=SYSDA,SPACE=(TRK,(5,5),RLSE),
//          DCB=(RECFM=VB,BLKSIZE=6000)
//SYSIN     DD *
*****
* Extract account # 34567-8 only *
*****
SYSUT1=DUMMY
COPYDIFF
MAXDIFF=1
FILTERIN=(9,EQ,X'0345678C')
```

Select Two Accounts - FILTORIN's

We want to send only those input records for accounts 123 and 234 to the comparison routines:

```
FILTORIN=(2,EQ,C'123')
FILTORIN=(2,EQ,C'234')
```

Exclusive Filters

We want to send to the comparison routines only those records where position 3 is 'A' and position 7 is 'X'. We want every record that does not pass both these criteria to be bypassed:

```
FILTERIN=(3,EQ,C'A')
FILTERIN=(7,EQ,C'X')
```

or

```
FILTERIN=(3,EQ,C'A...X')
```

Filter Out One Record

We want to send all the input records to the compare routines except records for account 789:

```
FILTEROUT=(2,EQ,C'789')
```

or

```
FILTERIN=(2,NE,C'789')
```

Filter Out All But Certain Records

We want to send all of the records to the compare routines except records for divisions 12 and 23:

```
FILTROUT=(22,EQ,C'12')  
FILTROUT=(22,EQ,C'23')
```

Filter Out and Filter In

We want to send only those records where the account balance is over \$100,000 to the compare routines:

```
FILTEROUT=(97,EQ,X'...D') /* Eliminate negative balances */  
FILTERIN=(93,GE,X'...1') /* Select balances over $100,000 */
```

Disregard Inserted Records

A special payroll change is expected to insert one or more records behind each existing master record, using the same account number. We want to compare the old master to the new master, disregarding the inserted records:

```
KEY=(1,9),PRINT=NOMISMATCH
```

or

```
KEY=(1,9),FILTEROUT=(10,GE,X'3C')
```

Complex Filtering

We want to send to the compare routines only those records where all these things are true:

- Account balance is not zero
- State code is Texas
- Last name starts with CW or KW or Q

```
FOUT=(71,EQ,X'0000000.') /* Eliminate zero balances */  
FILTERIN=(19,EQ,C'TX') /* Select Texans */
```

```

FILTORIN=(21,EQ,C'CW')      /* Select 'CW' */
FILTORIN=(21,EQ,C'KW')      /* Select 'KW' */
FILTORIN=(21,EQ,C'Q')       /* Select 'Q' */

```

IDENTITY's and FIELD's

We want to compare files that have two different record types. Account header records have the letter 'A' in position 5 and we want to compare positions 17 through 29. Account detail records have the letter 'D' in position 5 and we want to compare positions 51 through 60:

```

KEY=(1,9,Z,A)      /* Zoned KEY - may have differing signs */
IDENTITY=(5,EQ,C'A') /* Identify Account Header record */
FIELD=(17,13)
IDENTITY=(5,EQ,C'D') /* Identify Detail record */
FIELD=(51,10)

```

COBOL Source Code Changes

The operations manager wants to send a list of changes to any accounts payable COBOL program to the Controller:

```
TEXT=COBOL
```

FILTER's With TEXT

We have just received the complete source code and execution JCL from a software vendor, and this is supposed to replace a prior release from six months before. Unfortunately, there is little documentation in what has changed in this release. Each piece of the release is uniquely named. The old version resides in a PANVALET library and the new version is in a large PDS. A naming convention pattern has been established, and it has been determined that all execution JCL procedures begin with the letter 'J' or 'XJ'. All COBOL modules are prefixed with 'PTXA'. There are a few PL/1 members that are suffixed with the letter 'P'. Control card members all start with 'C' and have a dollar sign in the fourth position. All the rest of the members are Assembler Language. We want to compare the new release with the prior release.

```

//COMPARE PROC
//COMPAREX EXEC PGM=COMPAREX
//STEPLIB DD DISP=SHR,DSN=somnode.COMPAREX.LOAD
//SYSPRINT DD SYSOUT=*
//PANDD1 DD DISP=SHR,DSN=somnode.PANLIB
//SYSUT2 DD DISP=SHR,DSN=somnode.PDS
//
// PEND
//*
//JCL EXEC COMPARE
SYSUT1=PAN
TEXT=JCL, FORIN=(MEMBER,1,EQ,C'J'),FIN=(M,1,EQ,C'XJ')
//COBOL EXEC COMPARE
SYSUT1=PAN
TEXT=COBOL,FIN=(M,1,EQ,C'PTXA'),FOROUT=(M,8,EQ,C'P')
//PL1 EXEC COMPARE
SYSUT1=PAN
TEXT=PL/1,FIN=(M,1,EQ,C'PTXA'),FIN=(M,8,EQ,C'P')
//CONTROL EXEC COMPARE

```

```

SYSUT1=PAN
TEXT=JCL,FIN=(M,1,EQ,C'C..$')
//BAL      EXEC COMPARE
      SYSUT1=PAN
TEXT=BAL,FOUT=(M,1,EQ,C'J'),FOUT=(M,1,EQ,C'XJ')
      FOUT=(M,1,EQ,C'PTXA'),FOUT=(M,1,EQ,C'C..$')

```

or

```

//ALLOFIT EXEC COMPARE
      SYSUT1=PAN
      WILDCARD=C'*,TEXT=*

```

Audit PDS Libraries

The auditors of a financial institution insist that all load modules within production libraries be accounted for with respect to source code. The shop standard is that Partitioned DataSets (PDS's) be used to hold source code, object code, and load modules, with the same name used across the libraries. We want to compare any two libraries:

```

DIRECTORY      /* Just compare directories
PRINT=NOMATCH  /* Print only the mismatches

```

Compare JCL Libraries

An analyst borrowed a library of JCL. It was created via ISPF/PDF (or TSO/SPF) originally. He copied it to his or her USER ID via IEBCOPY and then changed it. We want to compare the two libraries to see what he changed:

```

//DIR$PDF EXEC COMPARE
DIR=PDF      /* Walk Directory, show members added      */
              /*                                         changed      */
              /*                                         or deleted. */
//*
//TEXT      EXEC COMPARE
TEXT=JCL     /* On matched Member names, see the      */
PRINT=FULL  /* Differences in Source Code in context. */

```

Regression Test in Database Environment

- We want to read the database direct through the COMPAREX interface (CPXIFACE) and write each record out to SYSUT3 which is flat.

```

SYSUT1=DUMMY
      /* Assume CPXIFACE Generated correctly */
SYSUT2=(OTH, MEMBER=DBDNAME)
COPYDIFF,MAXDIFF=5,CONTINUE=YES /* Write it all to SYSUT3 */

```

Compare to Backup

- After the database has been updated by our application programs, we want to compare it against the backup (just created) to see the changes made.

```

SYSUT2=(OTH, MEMBER=DBDNAME)      /* Read Database again */
MAXDIFF=50, CONTINUE               /* Generally advised */
SEGMENT=(1, EQ, C'ROOT', (A, 9, 5)) /* ROOT Segment */
  IDENTITY=(1, EQ, C'ROOT')
  FIELD=(65, END)
  MASK=(98, 3)
SEGMENT=(1, EQ, C'APPLES')
  ID=(1, EQ, C'APPLES')
  FIELD=(65, END)
SEG=(1, EQ, C'STEMS', (R, 14, 2))
  ID=(1, EQ, C'STEMS')
  FIELD=(65, END)
  MASK=(77, 1)
  MASK=(81, 1)

```

Find Latest Versions

At least two programmer/analysts have copied executable load modules from the same load library into their own load libraries for special concatenation of Joblib/Steplib. Then, certain modules have been recompiled and linkage edited into some of the libraries. Now, nobody really knows what version is the latest and the multiple copies are wasting disk storage space.

We want to compare any two of these libraries (A to B, B to C, and so forth) to isolate the newest version of a module so we can scratch the old versions. There is a date/time stamp generated by the compiler and other date/time stamp generated by the linkage editor in each load module. Assuming that the compiler and linkage editor have not changed between versions of any module, the only changes should be the modified source code and the date of compilation.

```

//A2B EXEC COMPARE
DATA=CSECT /* CSECT Parsing */
MAXDIFF=5, CONTINUE
MBRHDR=COND /* Only see the members that differ */
//B2C EXEC COMPARE
DATA=CSECT /* CSECT Parsing */
MAXDIFF=5, CONTINUE
MBRHDR=COND /* Only see the members that differ */

```

IEBUPDTE Formatting of Audit Trail

A shop has neither PANVALET nor LIBRARIAN. All source code updates are done through IEBUPDTE. We want to create an audit trail of changes between two PDS's, then have IEBUPDTE recreate the first PDS.

```

//AUDIT EXEC PGM=COMPAREX, REGION=256K
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR, DSN=somnode.PDS1.ASSEMBLE
//SYSUT2 DD DISP=SHR, DSN=somnode.PDS2.ASSEMBLE
//SYSUT3 DD DISP=(, CATLG, DELETE), DSN=somnode.IEBUPDTE.AUDIT,
// UNIT=sysda, SPACE=(TRK, (1, 5), RLSE),
// DCB=(RECFM=FB, LRECL=80, BLKSIZE=2000)
//SYSIN DD *
TEXT=BAL /* Assume both libraries contain Assembler Source */
MBRHDR=COND /* Only see the members that have indeed changed */

```



```

* Create Audit Trail in IEBUPDTE Format *
  COPYDIFF=(IEBUPDTE,SEQFLD='738,765') /* <=== NOTE SEQFLD=.. */
/*
//CPXUPDTE EXEC PGM=IEBUPDTE,PARM=MOD
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DSN=somnode.PDS1.ASSEMBLE
//SYSUT2 DD DISP=SHR,DSN=somnode.PDS1.ASSEMBLE <=== Same Name
//SYSIN DD DISP=SHR,DSN=somnode.IEBUPDTE.AUDIT
/*
//VERIFY EXEC PGM=COMPAREX,REGION=256K
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DSN=somnode.PDS1.ASSEMBLE
//SYSUT2 DD DISP=SHR,DSN=somnode.PDS2.ASSEMBLE
//SYSIN DD *
  TEXT=BAL /* Verify that Matching Members are Identical */
  MBRHDR=COND /* This will make for a very short report */
/* EOJ

```

Desensitize Live Production Data

Instead of creating special test data to test out enhanced modules, a shop uses live production files to test their changes on. The internal auditor has insisted that live names, addresses and any other sensitive information be "clobbered" before comparison and/or printing.

```

MAXDIFF=50,CONTINUE /* Generally advised */
IDENTITY=(21,EQ,C'A')
  DESEN=(35,C'OBLITERATE THE NAME FIELD ')
  DESEN=(65,C'OBLITERATE THE FIRST ADDRESS ')
  MASK=(111,5) /* Date Time Stamp */
IDENTITY=(21,EQ,C'B')
  /* Average Balance over 3 years */
  DESEN=(31,X'0000000C')

```

Reverse Delta Deck

It is very common to have very many versions of the source code for any one program. We can create an audit trail of the changes by comparing the old version against the newer version at each change level. However, we can also compare them in reverse order and create a delta deck such that if the old version of the source code is lost, it can be recreated by running the delta deck against the new version.

This concept can be used to save disk space for older versions of source code. Only the proper delta decks need be saved.

We will assume that the new version (SYSUT1) resides in a PDS and the old version (SYSUT2) resides in a PANVALET library. We will create a delta deck (SYSUT3) for module "PROGRAM1" and save it in another PDS.

```

//COMPAREX EXEC PGM=COMPAREX
//STEPLIB DD DISP=SHR,DSN=somnode.COMPAREX.LOAD
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DSN=somnode.SOURCE.PDS <=== New version
//SYSUT2 DD DISP=SHR,DSN=somnode.PANLIB <=== Old version
//SYSUT3 DD DISP=SHR,DSN=somnode.DELTA.PDS(PROGRAM1)

```

```
//SYSIN      DD *  
SYSUT2=(PAN,DDNAME=SYSUT2),TEXT=COBOL  
FIN=(MEMBER,1,EQ,C'PROGRAM1')  
      /* Create Delta Deck in PANVALET Format */  
COPYDIFF=(PAN,STAMP=YES) /* Time stamp when created */  
/* EOI
```

Appendix A. Sample COBOL Programs

COBOL1 - Before Change

000100 IDENTIFICATION DIVISION.	00000100
000200 PROGRAM-ID. COBOL01.	00000200
000300 ENVIRONMENT DIVISION.	00000300
000400 INPUT-OUTPUT SECTION.	00000400
000500 FILE-CONTROL.	00000500
000600 SELECT ONLY-FILE,	00000600
000700 ASSIGN VSAMFILE,	00000700
000800 ORGANIZATION IS INDEXED,	00000800
000900 ACCESS DYNAMIC,	00000900
001000 RECORD KEY IS ONLY-KEY,	00001000
001100 FILE STATUS IS ONLY-FILE-STAT.	00001100
001200 DATA DIVISION.	00001200
001300 FILE SECTION.	00001300
001400 FD ONLY-FILE.	00001400
001500 01 ONLY-REC.	00001500
001600 02 ONLY-KEY.	00001600
001700 03 ONLY-ACCOUNT PIC X(10).	00001700
001800 03 ONLY-TYPE PIC XX.	00001800
001900 03 ONLY-DSN PIC X(44) OCCURS 2.	00001900
002000 03 ONLY-MEMBER PIC X(10) OCCURS 2.	00002000
002100 02 ONLY-REST-OF-REC PIC X(100).	00002100
002200 WORKING-STORAGE SECTION.	00002200
002300 77 ONLY-FILE-STAT PIC XX.	00002300
002400 01 SWITCHES.	00002400
002500 02 END-OF-ONLY-FILE-SW PIC X.	00002500
002600 88 END-OF-ONLY-FILE VALUE 'Y'.	00002600
002700 LINKAGE SECTION.	00002700
002800 01 LS-FUNCTION PIC X(8).	00002800
002900 88 OPEN-REQUEST VALUE 'OPEN'.	00002900
003000 88 READSEQ-REQUEST VALUE 'READSEQ'.	00003000
003100 88 CLOSE-REQUEST VALUE 'CLOSE'.	00003100
003200 01 LS-ONLY-REC PIC X(220).	00003200
003300 EJECT	00003300
003400 PROCEDURE DIVISION USING LS-FUNCTION, LS-ONLY-REC.	00003400
003500 MAIN-LINE.	00003500
003600 IF OPEN-REQUEST PERFORM DO-THE-OPEN	00003600
003700 ELSE IF READSEQ-REQUEST PERFORM DO-THE-SEQ-READ	00003700
003800 ELSE IF UPDATE-REQUEST PERFORM DO-THE-UPDATE	00003800
003900 ELSE IF CLOSE-REQUEST PERFORM DO-THE-CLOSE	00003900
004000 ELSE DISPLAY 'INVALID I/O FUNCTION REQUESTED'	00004000
004100 MOVE 12 TO RETURN-CODE.	00004100
004200 GOBACK.	00004200
004300 DO-THE-OPEN.	00004300
004400 OPEN I-O ONLY-FILE.	00004400
004500 IF ONLY-FILE-STAT = '00'	00004500
004600 MOVE 0 TO RETURN-CODE	00004600
004700 ELSE	00004700
004800 EXHIBIT NAMED ONLY-FILE-STAT	00004800
004900 DISPLAY 'OPEN FAILED'	00004900
005000 MOVE 8 TO RETURN-CODE.	00005000
005100 DO-THE-SEQ-READ.	00005100
005200 READ ONLY-FILE NEXT, AT END MOVE 8 TO RETURN-CODE.	00005200
005300 IF ONLY-FILE-STAT = '00'	00005300
005400 MOVE ONLY-REC TO LS-ONLY-REC	00005400
005500 MOVE 'N' TO END-OF-ONLY-FILE-SW	00005500
005600 ELSE	00005600
005700 MOVE 'Y' TO END-OF-ONLY-FILE-SW	00005700
005800 MOVE 8 TO RETURN-CODE.	00005800
005900 DO-THE-CLOSE.	00005900
006000 CLOSE ONLY-FILE.	00006000

Figure 64. COBOL1 Before Change

COBOL1 - After Change

000100 IDENTIFICATION DIVISION.	00000100
000200 PROGRAM-ID. COBOL1.	00000200
000300 ENVIRONMENT DIVISION.	00000300
000400 INPUT-OUTPUT SECTION.	00000400
000500 FILE-CONTROL.	00000500
000600 SELECT ONLY-FILE,	00000600
000700 ASSIGN VSAMFILE,	00000700
000800 ORGANIZATION IS INDEXED,	00000800
000900 ACCESS DYNAMIC,	00000900
001000 RECORD KEY IS ONLY-KEY,	00001000
001100 FILE STATUS IS ONLY-FILE-STAT.	00001100
001200 DATA DIVISION.	00001200
001300 FILE SECTION.	00001300
001400 FD ONLY-FILE.	00001400
001500 01 ONLY-REC.	00001500
001600 02 ONLY-KEY.	00001600
001700 03 ONLY-ACCOUNT PIC X(10).	00001700
001800 03 ONLY-TYPE PIC XX.	00001800
001900 03 ONLY-DSN PIC X(44) OCCURS 2.	00001900
002000 03 ONLY-MEMBER PIC X(10) OCCURS 2.	00002000
002100 02 ONLY-REST-OF-REC.	00002100
002200 05 ONLY-DISP PIC XXX.	00002200
002300 05 ONLY-UNIT PIC X(8).	00002300
002400 05 ONLY-VOL PIC X(6).	00002400
002500 05 FILLER PIC X(83).	00002500
002600 WORKING-STORAGE SECTION.	00002600
002700 77 ONLY-FILE-STAT PIC XX.	00002700
002800 01 SWITCHES.	00002800
002900 02 END-OF-ONLY-FILE-SW PIC X.	00002900
003000 88 END-OF-ONLY-FILE VALUE 'Y'.	00003000
003100 LINKAGE SECTION.	00003100
003200 01 LS-FUNCTION PIC X(8).	00003200
003300 88 OPEN-REQUEST VALUE 'OPEN'.	00003300
003400 88 READSEQ-REQUEST VALUE 'READSEQ'.	00003400
003500 88 CLOSE-REQUEST VALUE 'CLOSE'.	00003500
003600 01 LS-ONLY-REC PIC X(220).	00003600
003700 EJECT	00003700
003800 PROCEDURE DIVISION USING LS-FUNCTION, LS-ONLY-REC.	00003800
003900 MAIN-LINE.	00003900
004000 IF OPEN-REQUEST PERFORM DO-THE-OPEN	00004000
004100 ELSE IF READSEQ-REQUEST PERFORM DO-THE-SEQ-READ	00004100
004200 ELSE IF CLOSE-REQUEST PERFORM DO-THE-CLOSE	00004200
004300 ELSE DISPLAY 'INVALID I/O FUNCTION REQUESTED'	00004300
004400 MOVE 12 TO RETURN-CODE.	00004400
004500 GOBACK.	00004500
004600 DO-THE-OPEN.	00004600
004700 OPEN I-O ONLY-FILE.	00004700
004800 IF ONLY-FILE-STAT = '00'	00004800
004900 MOVE ZERO TO RETURN-CODE	00004900
005000 ELSE	00005000
005100 EXHIBIT NAMED ONLY-FILE-STAT	00005100
005200 DISPLAY 'OPEN FAILED'	00005200
005300 MOVE 8 TO RETURN-CODE.	00005300
005400 DO-THE-SEQ-READ.	00005400
005500 READ ONLY-FILE NEXT, AT END MOVE 8 TO RETURN-CODE.	00005500
005600 IF ONLY-FILE-STAT = '00'	00005600
005700 MOVE ONLY-REC TO LS-ONLY-REC	00005700
005800 MOVE 'N' TO END-OF-ONLY-FILE-SW	00005800
005900 ELSE	00005900
006000 MOVE 'Y' TO END-OF-ONLY-FILE-SW	00006000
006100 MOVE 8 TO RETURN-CODE.	00006100
006200 DO-THE-CLOSE.	00006200
006300 CLOSE ONLY-FILE.	00006300

Figure 65. COBOL1 After Change

Glossary

ACB	Access Control Block. A term describing the attributes of a VSAM file to be read or written.		
BASELINE	a set of data and software that is known to produce correct or at least acceptable results. See TESTLINE.		
BINARY DECIMAL	the format to internally store numeric fields where the value has been converted from decimal to binary and the sign is in the leftmost bit of the field. == => X'A03F' = Decimal -9255.		
BMP	Batch Message Program. A term used in IMS to describe a program running under IMS that is batch oriented (that is, not on-line).		
BUFF	a keyword that specifies the buffer size for TEXT comparison logic or DATA logic with Random keys.		
CINV	an abbreviation for Control Interval. A control interval is a VSAM term roughly the equivalent of BLKSIZE.		
COMPAREX	the Comparison utility. Pronounced COMP (rhymes with POMP) A (rhymes with HUH) REX.		
CONTINUE	a keyword that causes the utility to read records and count differences without printing differing records. Used with MAXDIFF.		
COPYDIFF	abbreviation for COPY the DIFFerences to a third file. A keyword that causes COMPAREX to write file SYSUT3.		
DATA	(1) files that have an inter-record relationship; (2) comparison logic based on synchronization of records; (3) a keyword to specify DATA comparison logic.		
CSECT	abbreviation for Control SECTION. Each executable module can be broken up into one or more CSECTs, the order and content of which are independent of other CSECTs.		
DBMS	DataBase Management System. Global term used to describe monitors whose charge is to coordinate the efforts of application programs to the data on large random access files (databases). Telecommunications support is usually supplied also.		
DCB	Data Control Block. A term describing the attributes of a non-VSAM file to be read or written.		
DELTA DECK	a dataset (used to be a card deck) containing the changes (Greek letter Delta) in transaction format such that if it is fed back into a predetermined library management system (PANVALET, LIBRARIAN), it will update the old dataset to be exactly like the new dataset.		
	DESENSITIZING the act of obliterating sensitive data fields within live production data files.		
	DIFFERENCES REPORT the major output from COMPAREX. The report shows records that differ and processing totals.		
	DIRECTORY a keyword to specify comparison of only the DIRECTORY portions of directory-embedded datasets. A series of 256-byte records at the beginning of a partitioned dataset (PDS) that contains an entry for each member in the dataset. Note that it is possible to physically read this special part of the PDS by specifying "SYSUT1=QSAM" while actually pointing to a PDS.		
	DUMMY In JCL, specifying DUMMY forces the associated program opening the dataset to go to end-of-file at the first read or to nullify every write instruction to the file.		
	ESDS Entry Sequenced DataSet. A VSAM term meaning that the sequence of the data within is determined by the order in which they are stored. Each new record is stored after the last record in the dataset.		
	FIELDS keywords that cause COMPAREX to compare only on specified positions of the input records.		
	FILTER a keyword that performs a logical test to select or reject records. Selected records are eligible for comparison.		
	FILTERING the process of selecting or rejecting records with the use of FILTERs.		
	FRAME a keyword, used with TEXT comparison logic, to specify the surrounding of differing blocks of records with the PLUS character and the DASH character on the DIFFERENCES REPORT.		
	GENFLDS (1) an optional part of the Differences Report that shows a visual representation of each input record type; (2) a keyword specifying these visual representations.		
	HELP a keyword that causes COMPAREX to display a listing of valid keywords and their definitions onto SYSPRINT.		
	IDCAMS Access Method Services. IBM's system of creating, deleting, printing, loading, and un-		

	loading files of almost any organization. Any VSAM use mandates the use of IDCAMS.	OS	Operating System. There are many variations and release levels. e.g. MVS, MVS/XA.
IDENTITY	a keyword that performs a logical test to identify a record type on the SYSUT1 file.	PACKED DECIMAL	the format to internally store numerics where each digit takes a half-byte (nibble or four bits) and the sign is at the end of the field. == => X'0123456C'.
IMS	Information Management System. IBM's major entry in the DBMS marketplace.	PDS	partitioned dataset - a collection of independent groups (called MEMBERS) of sequential records on a direct access volume. Each MEMBER has a unique name and is listed in a DIRECTORY at the beginning of the dataset. It is a directory-embedded dataset.
ISAM	Indexed Sequential Access Method.	QSAM	Queued Sequential Access Method - also Flat File.
JCL	job control language - statements which describe a job to the operating system and inform the system of how a job is to be processed.	RDW	Record Descriptor Word. This word (four bytes) describes the length of a record if it is variable length QSAM or ISAM. Sometimes referred to as the LLBB.
KEY	(1) a control field on a file; (2) a keyword specifying this control field.	RKP	Relative Key Position. The relative (to zero) position of a synchronizing key within a record. Usually used in conjunction with KEYLEN.
KEYLEN	the length of a synchronizing key. Usually used in conjunction with RKP.	RRDS	Relative Record DataSet. A VSAM term meaning that the file has no index. It is a string of fixed-length slots, each of which is identified by a relative record number from 1 to nnn, where nnn is the maximum number of records that can be stored in the dataset.
KEYWORD	an instruction given to COMPAREX to cause the utility to modify its default processing.	SEGMENT	(1) a piece of a database; (2) an identifier on a record to delineate at what level of a database it comes from; (3) a keyword to COMPAREX that allows synchronizing between two versions of a database.
KSDS	Key Sequenced DataSet. A VSAM term meaning that the sequence of the data within is determined by an indexing key structure. KSDS files may be accessed sequentially or randomly (by the key).	SKIPUT's	keywords that cause COMPAREX to bypass initial records on the input files.
MASKS	keywords that cause COMPAREX to ignore specified positions on the input records when comparing them.	SQUEEZE	(1) the removal of certain characters before TEXT comparison; (2) a keyword to specify a character to be removed.
MAXDIFF	a keyword to specify the maximum number of differences to be shown on the Differences Report.	STOPAFT	a keyword that specifies the maximum number of records to be read from either input file.
MEMBER	an independent, sequentially organized dataset identified by a unique name in a directory-embedded dataset.	SYNCHRONIZATION	the pairing of records for comparison by COMPAREX. KEY, SEGMENT, and same physical-record number synchronization are supported with DATA comparison logic.
MLC	matching line count. A keyword to specify the number of exact matches COMPAREX makes before deciding that TEXT comparison logic is back into synchronization.	SYSIN	the input file containing keywords.
MODE	a keyword to specify the user's orientation. MODE=APPLICATIONS causes displacements in other keywords to be relative to one; MODE=SYSTEMS causes displacements to be relative to zero.	SYSPRINT	the output file from COMPAREX that contains the Differences Report.
MPP	Message Processing Program. A term in IMS to describe an online program that is driven by input messages, usually from a terminal operator.		
NIBBLE	a half byte. Four bits of data.		
NULLFILE	specifying DSN=NULLFILE is the same as specifying DD DUMMY (see DUMMY).		

SYSUT1	the first of the two input files to be compared. SYSUT1 is usually the old master or the unmodified file.		based on record-to-record matching; (3) the keyword to specify TEXT comparison logic.
SYSUT2	the second of the two input files to be compared. SYSUT2 is usually the new master or the modified file.	VSAM	Virtual Storage Access Method.
SYSUT3	an optional output file from COMPAREX, written when COPYDIFF is specified.	WILDCARD	a keyword used to specify a character to be used in other keywords to indicate that any value passes logical tests specified.
TESTLINE	a set of data and software that is not yet ready to be placed into production mode. See BASELINE.	ZONED DECIMAL	the format to display numerics where each digit is preceded by its sign and takes a whole byte. = = = > X'F1F5C0'.
TEXT	(1) files that have no inter-record relationship; (2) comparison logic that pairs records		

Index

- DEL
 - with DELETE
 - message CPX15I 157
- INS
 - with INSERT
 - message CPX15I 157
- REP
 - with REPLACE
 - message CPX15I 157
- SEL 158

A

- A - Ascending
 - message CPX19I 160
 - with KEY 108
 - message CPX07I 152
 - with KEY1 108
 - with Keys 152
 - with SEGMENT 110
- abend 8, 85, 108, 127, 160, 178
 - 80A 31, 178
 - S0C1 through S0CF 178
 - S0C3 131
 - S80A 31, 178
- abnormal end 178
- ACB
 - message CPX16I 159
 - message CPX21I 161
 - message CPX22I 161
- ADABAS 32
- ADABAS Interface Record Layout 32
- ALC
 - with TEXT 122
- ALL
 - with HELP 139
- ALPHA
 - with LINE 142
 - message CPX08I 153
- Alphanumeric Format - FORMAT = 1y 138
- APL
 - with MODE 94
- APPLICATIONS
 - with MODE 94
 - message CPX06I 151
- archie 96
- ASCII
 - keyword definition 133
 - message CPX08I 153
- audit trail 28, 185

B

- B - Binary
 - with FIELD 85
 - with FIELD1 85
 - with FIELD2 86
 - with KEY 108
 - message CPX07I 152
 - with KEY1 108
 - with KEY2 109
 - with MASK 92
 - with MASK1 92
 - with MASK2 92
- BACKUP 96
- BAL 182
 - with TEXT 122
- BLKSIZE
 - message CPX16I 159
 - message CPX21I 161
 - message CPX22I 161
- BUFF 114
 - keyword definition 120
 - message CPX25I 163
 - message CPX39A 166
- bytes underscored
 - message CPX74I 173

C

- C
 - with DELETE
 - message CPX15I 157
 - with INSERT
 - message CPX15I 157
 - with REPLACE
 - message CPX15I 157
 - with TEXT 122
- C - Character
 - with FIELD 85
 - with FIELD1 85
 - with FIELD2 86
 - with KEY 108
 - message CPX07I 152
 - with KEY1 108
 - with KEY2 109
 - with MASK 92
 - with MASK1 92
 - with MASK2 92
- CASE
 - keyword definition 133
 - message CPX08I 153
- CINV
 - message CPX16I 159
 - message CPX21I 161

- message CPX22I 161
- CLIST 54
 - with TEXT 122
- CLOS 160
- COBOL 182
 - with TEXT 48, 49, 122
 - example 28, 29
- COBOL1 After Change 188
- COBOL1 Before Change 187
- comments 12, 26, 130, 148
 - example 70
- COMPAREX General Flowchart 8
- COMPAREX interface
 - message CPX93A 178
 - message CPX94A 178
- compatible
 - file organization 164
- concatenated key 36, 110
- COND 184
 - with HALT 11, 138, 148
 - installation defaults 9
 - message CPX05I 151
 - message CPX30A 164
 - with MBRHDR 143
 - message CPX05I 151
- condition code 11
- context 123
- CONTINUE 150, 171, 183, 184
 - keyword definition 78
- COPYDIFF 8, 13, 180, 183-185
 - keyword definition 126
 - message CPX15I 157
- CPXSISPF 54
- CPX00I 12, 148
- CPX01I 12, 149
- CPX02A 149
- CPX03I 150
- CPX04I 150
- CPX05I 151
- CPX06I 151
- CPX07I 152
- CPX08I 153
- CPX09I 154
- CPX10I 154
- CPX11I 155
- CPX12I 155
- CPX13I 157
- CPX15I 157
- CPX16A 159
- CPX16I 12, 159
- CPX18I 102, 160
- CPX19I 160
- CPX20I 28, 160
- CPX21I 12
- CPX22I 12, 161
- CPX23I 40, 162
- CPX24A 162
- CPX25I 12, 163
- CPX26I 163
- CPX27I 164
- CPX30A 148, 164
- CPX31A 31, 40, 49, 138, 164
- CPX35A 100, 165
- CPX36A 100, 102, 165
- CPX37A 102, 165
- CPX39A 166
- CPX40A 166
- CPX41I 166
- CPX42I 167
- CPX51I 101, 168
 - example 11
- CPX52I 101, 168
 - example 11
- CPX56I 169
- CPX57I 169
- CPX61I 100, 170
- CPX62I 100, 170
- CPX64I 170
- CPX65I 170
- CPX66A 171
- CPX67I 171
 - example 11
- CPX69I 171
- CPX71I 172
 - example 11
- CPX72I 172
 - example 11
- CPX74I 107, 173
 - example 11
- CPX75I 107, 173
 - example 11
- CPX76I 107, 174
- CPX77I 174
- CPX78I 107, 143, 175
- CPX80I 175
 - condition code 11
 - example 11
- CPX90A 11, 176
- CPX91A 176
- CPX92A 177
- CPX93A 178
- CPX94A 178
- CPX97A 178
- CPX99A 120, 179
- CPX@PRIM 55
- CPXIFACE 7
 - INFO call 28
 - keyword definition 78
 - message CPX20I 160
 - message CPX93A 178
 - message CPX94A 178
- CPXIFACE Sample Code - Roll Your Own Slot 51
- CPXJOBST 64
- CPXOPDAT 57
- CPXOPDS1 59, 65
- CPXOPDS2 60, 66
- CPXOPDSN 65
- CPXOPDTO 56
- CPXOPSYK 58
- CPXPRLOD 68

- CPXPRSAV 67
- CPXREPR 62
- CPXRUNPR 61
- CPXSHORT 63
- CSECT 79, 189
 - Parsing 79
 - with DATA 79, 166-168
 - with FILTERIN 87
 - with FILTEROUT 88
 - with filters
 - message CPX10I 154
 - with FILTERIN 88
 - with FILTEROUT 89
- CSECT Parsing 120
 - with filters 87
- Csect Parsing 163
- csect parsing 166-168
- CSECT Parsing - Execution JCL and Keywords 81
- CSECT Parsing - Sample Difference Report 81
- current management code 96
- CYCLE 96

D

- D - Descending
 - message CPX19I 160
 - with KEY 108
 - message CPX07I 152
 - with KEY1 108
 - with Keys 152
 - with SEGMENT 110
- DA
 - DSORG
 - message CPX16I 159
 - message CPX21I 161
 - message CPX22I 161
- DASH
 - count of 173
 - keyword definition 134
 - message CPX11I 155
- DATA 173
 - CSECT Parsing 79
 - end of 172
 - keyword definition 79
 - message CPX24A 162
 - message CPX25I 163
 - message CPX39A 166
 - with HELP 139
- DATA Example - Modified to Fit Page 11
- Data File Synchronization Properties 99
- DATA=CSECT
 - message CPX25I 163
- Database Structure - Random Key Synchrony 53
- DBD 37, 38
- DCB
 - message CPX16I 159
 - message CPX21I 161
 - message CPX22I 161
- DDNAME 71

- with SYSUT1 49, 96
- with SYSUT2 97
 - example 29
- DECIMAL 141
 - keyword definition 134
 - message CPX08I 153
- DEL
 - with DELETE
 - message CPX15I 157
- DELETE
 - keyword definition 128
- delta deck 126, 157, 185
 - definition 189
- DESEN 185
 - keyword definition 82
 - message CPX12I 155
 - unusable
 - message CPX76I 174
- DESEN1
 - keyword definition 82
 - message CPX12I 155
- DESEN2
 - keyword definition 82
 - message CPX12I 155
- Desensitizing data 185
- DFSRR00 37, 38
- differences report 6, 8, 11, 12, 14, 17-20, 27, 130
- differing members
 - message CPX78I 175
- differing records
 - message CPX75I 173
- DIR 83, 183
- Direct ADABAS Interface 32
- Direct CAMLIB Interface 35
- Direct DL/1 Interface 37, 38
- Direct DMS Interface 39
- Direct GEM Interface 30
- Direct IAM Interface 40
- Direct IDMS Interface 41
- Direct IDMS Interface Via Exit Module - Compare 44
- Direct IDMS Interface Via Exit Module - Unload 43
- Direct ISPF Packed File Interface 46
- Direct LIBRARIAN Interface 29, 30
- Direct PANVALET Interface 28, 29
- Direct RAMIS Interface 47
- Direct ROSCOE Interface 48
- Direct WYLBUR Interface 49
- DIRECTORY 183
 - end of 172
 - keyword definition 83
 - message CPX25I 163
- directory-embedded 175
- DISPLAY
 - with HELP 139
- Display Keyword Properties 130
- Ditto Format
 - See LINE 142
- ditto format 136, 142
- DL/1 36, 52
 - DBD 37, 38

- example 37, 38
- FASTPATH 37
- PSB 37, 38
- DL/1 Interface Segment Layout 36
- DSORG 129
 - message CPX16I 159
 - message CPX21I 161
 - message CPX22I 161
- DUMMY 183
 - with SYSUT1 96
 - with SYSUT2 97
 - with SYSUT3 129
- Dump Format
 - See LINE 142

E

- EBCDIC
 - keyword definition 135
 - message CPX08I 153
- EMOD
 - with COPYDIFF=LIB 158
- END
 - with COPYDIFF=LIB 158
 - with FIELD 85
 - with FIELD1 85
 - with MASK 92
 - with MASK1 92
- end of job 175
- EQ 73, 87
- ERROR? 12, 27, 70, 132, 138, 148, 164
- ESD 166
- ESDS
 - message CPX16I 159
 - message CPX21I 161
 - message CPX22I 161
- execution JCL 25
- expected changes 6
- expired
 - license 179
- External Symbol Dictionary 166
- extra record 169

F

- F - FORMAT abbreviation 136
- FASTPATH 36, 37
- FIELD 138, 182, 183
 - keyword definition 85
 - message CPX12I 155
 - message CPX52I 168
 - unusable
 - message CPX76I 174
- FIELD1
 - keyword definition 85
 - message CPX12I 155
- FIELD2
 - keyword definition 86

- message CPX12I 155
- file organization 164
- Filter 174
- FILTERIN 72, 180, 181
 - keyword definition 87
 - message CPX10I 154
- filtering 87-89
 - CSECT 87-89
 - members rejected
 - message CPX78I 175
 - records rejected
 - message CPX77I 174
 - short records 165
 - statistics
 - message CPX75I 173
 - unusable
 - message CPX77I 174
- FILTEROUT 72, 181
 - keyword definition 88
 - message CPX10I 154
- FILTORIN 72, 180, 181
 - keyword definition 88
 - message CPX10I 154
- FILTOROUT 72
 - keyword definition 89
 - message CPX10I 154
- FIN 72, 87, 182
- flat file 18, 20
- FLDSONLY
 - keyword definition 135
 - message CPX11I 155
- FORIN 72, 88, 182
- FORMAT
 - explanation
 - message CPX25I 163
 - keyword definition 136
 - message CPX25I 163
- FORMAT=02
 - DECIMAL
 - NIBBLE (Excerpts) 135
- FOROUT 72, 89
- FORTRAN
 - with TEXT 122
- FOUT 72, 88
- FRAME
 - keyword definition 121
 - message CPX25I 163
- free-form keywords 26
- FULL 183
 - with PRINT 123, 147
 - message CPX05I 151

G

- GE 73, 87
- GEM 7
 - example 30
 - with COPYDIFF 126
 - message CPX15I 157

GENFLDS
 keyword definition 138
 message CPX13I 157
GT 73, 87

H

HALT 11, 71, 148
 installation defaults 9
 keyword definition 138
 message CPX05I 151
 message CPX30A 164
HDAM 36
HELP 12, 70
 keyword definition 139
 messages 149
Help Key 55
HEX
 keyword definition 141
 message CPX08I 153
HIDAM 36
Hierarchical Database Structure 111
Hierarchical Database Structure - Segment Synchrony 52
HISAM 36
HOR
 with LINE 142
HORIZONTAL
 with LINE 142
 message CPX08I 153

I

IAM 40
 example 40
ID
 see IDENTITY 90
IDENTITY 138, 182, 183, 185
 FIELD
 MASK
 and DESEN Messages 91
 keyword definition 90
 message CPX12I 155
 message CPX52I 168
 unusable
 message CPX76I 174
IDMS 41
 example 41
 IDMS/R 41
IDMS COBOL Exit Module JOSEFINE - Excerpts 45
IDMS Interface Record Layout 41
IEBUPDTE 184
 with COPYDIFF 126
 message CPX15I 157
Ignore sign differences 141
IGNORSIN
 keyword definition 91, 141
ILV
 INTERLEAVE abbreviation 142
 image copy 36
 implicit KEY
 CPX23I 162
 -INC 96
 INCLUDE 96
 with SYSUT1 96
 with SYSUT2 97
 INFO 28, 160
 INPUT
 with HELP 139
 Input Keyword Properties 69
 -INS
 with INSERT
 message CPX15I 157
 INSERT
 keyword definition 128
 inserted members
 message CPX78I 175
 inserted records
 message CPX75I 173
 installation default 151
 Installation Defaults 9
 installation defaults 9
 INTERLEAVE
 keyword definition 142
 message CPX25I 163
IS
 DSORG
 message CPX16I 159
 message CPX21I 161
 message CPX22I 161
ISAM 7, 31, 36
 with SYSUT1 97
 with SYSUT2 97
 with SYSUT3 129
ISPF
 packed file 46
 example 46
ISPF Interface 28, 54
ISPF Panel CPX@PRIM 55
ISPF Panel CPXJOBST 64
ISPF Panel CPXOPDAT 57
ISPF Panel CPXOPDS1 59
ISPF Panel CPXOPDS2 60
ISPF Panel CPXOPDSN 65
ISPF Panel CPXOPDPTO 56
ISPF Panel CPXOPSYK 58
ISPF Panel CPXPRLOD 68
ISPF Panel CPXPRSAV 67
ISPF Panel CPXREPRT 62
ISPF Panel CPXRUNPR 61
ISPF Panel CPXSHORT 63
ISPF Panel VENDOR - Raw Form 54
ISPF Panels 54

J

JCL 182, 183
 execution 25

with TEXT 122

K

KEY 18, 20, 31, 181, 182
 example of Random 37, 47
 keyword definition 108
 message CPX07I 152
 message CPX18I 160
 message CPX35A 165
 message CPX36A 165
KEY synchronization
 message CPX61I 170
 message CPX62I 170
KEY1
 keyword definition 108
 message CPX07I 152
KEY2
 keyword definition 109
 message CPX07I 152
KEYLEN
 message CPX16I 159
 message CPX21I 161
 message CPX22I 161
KEYONLY
 keyword definition 142
 message CPX05I 151
KILLRC
 keyword definition 142
KILLSPIE 178
KSDS
 message CPX16I 159
 message CPX21I 161
 message CPX22I 161

L

LE 73, 87
LEVEL
 with SYSUT1 96
 with SYSUT2 97
LIB
 with COPYDIFF 126
 message CPX15I 157
 with SYSUT1 96
 with SYSUT2 97
LIBRARIAN 7
 example 29
 PARM 96
license
 expired 179
LINE 142
 keyword definition 142
 message CPX08I 153
LIST=ALL
 IEBUPDTE parameter 158
load module compare 79
logical error

VSAM

 message CPX91A 176

LOWER

 with CASE 133

LRECL

 message CPX16I 159
 message CPX21I 161
 message CPX22I 161

LT 73, 87

M

M

 MEMBER abbreviation 49

 with FILTERIN 87

 with FILTEROUT 88

 with FILTORIN 88

 with FILTOROUT 89

 with SYSUT1 96

 with SYSUT2 97

MASK 138, 183, 185

 keyword definition 92

 message CPX12I 155

MASK1

 keyword definition 92

 message CPX12I 155

MASK2

 keyword definition 92

 message CPX12I 155

MASTER 96

MASTIN 96

MATCH

 with MBRHDR 143

 message CPX05I 151

 with PRINT 147

 message CPX05I 151

MAXDIFF 150, 171, 180, 183, 184

 keyword definition 143

MBRHDR 184

 keyword definition 143

 message CPX05I 151

MEMBER 183

 with COPYDIFF 126

 message CPX15I 157

 with FILTERIN 87

 with FILTEROUT 88

 with filters

 message CPX10I 154

 with FILTORIN 88

 with FILTOROUT 89

 with SYSUT1 96

 with SYSUT1/SYSUT2 46, 48, 49

 with SYSUT2 97

Messages Issued at End of Job 11

MISMATCH

 KEY synchronization

 message CPX61I 170

 message CPX62I 170

 SEGMENT synchronization

- message CPX64I 170
- message CPX65I 170
- with PRINT 147
- message CPX05I 151
- MIXED
 - with CASE 133
 - message CPX08I 153
- MLC 114
 - keyword definition 121
 - message CPX25I 163
- MODE
 - keyword definition 94
 - message CPX06I 151
- MVS Nucleus 79

N

- Natural (ADABAS) 32
- NE 73, 87
- NIBBLE
 - keyword definition 144
 - message CPX11I 155
- NO
 - with FRAME 121
 - message CPX25I 163
 - with HALT 138, 148
 - message CPX05I 151
 - with INCLUDE 96, 97
 - with MBRHDR 143
 - message CPX05I 151
 - with PASS 126
 - message CPX15I 157
 - with RESEQ 126
 - message CPX15I 157
 - with STAMP 126
 - message CPX15I 157
 - with TEMP 126
 - message CPX15I 157
 - with VERS 126
 - message CPX15I 157
- NOMATCH
 - with PRINT 147
 - message CPX05I 151
- NOMISMATCH 142, 181
 - with PRINT 147
 - message CPX05I 151
- NUM
 - renumbering under GEM 128
 - with FRAME 121
 - message CPX25I 163

O

- ??
- OTH with COPYDIFF 158
- object code
 - Natural 33
- OPEN 160

- Option == => 55
- OTH 183
 - with COPYDIFF 126
 - message CPX15I 157
 - with SYSUT1 96
 - with SYSUT2 97
- OUTPUT
 - with HELP 139
- Output Keyword Properties 124

P

- P - Packed
 - with FIELD 85
 - with FIELD1 85
 - with FIELD2 86
 - with KEY 108
 - message CPX07I 152
 - with KEY1 108
 - with KEY2 109
 - with MASK 92
 - with MASK1 92
 - with MASK2 92
- PAGE
 - keyword definition 144
 - message CPX08I 153
- PAN 182
 - with COPYDIFF 126
 - example 28
 - message CPX15I 157
 - with SYSUT1 96
 - with SYSUT1/SYSUT2
 - example 28
 - with SYSUT2 97
- PAN#1 28
- PANDD1 96
- PANDD3 96
- PANDD4 96
- PANEL
 - with TEXT 122
- Panels 54
 - CPX@PRIM 55
 - CPXJOBST 64
 - CPXOPDAT 57
 - CPXOPDS1 59, 65
 - CPXOPDS2 60, 66
 - CPXOPDSN 65
 - CPXOPDTO 56
 - CPXOPSYK 58
 - CPXPRLOD 68
 - CPXPRSAV 67
 - CPXREPR 62
 - CPXRUNPR 61
 - CPXSHORT 63
 - VENDOR 54
- PANVALET 7
 - example 28, 29
- PARM
 - with IDMS 41

- with ROSCOE 48
- with SYSUT1 96
- with SYSUT2 97
- PASCAL
 - with TEXT 46, 122
- PASS
 - with COPYDIFF 126
 - message CPX15I 157
- PASSWORD
 - message CPX16I 159
 - message CPX21I 161
 - message CPX22I 161
 - with SYSUT1 97
 - with SYSUT2 97
 - with SYSUT3 129
- PDF 183
 - with DIRECTORY 83
 - message CPX25I 163
- PDS
 - with SYSUT1 97
 - with SYSUT2 97
- PL/1 182
 - with TEXT 122
- PL/I
 - with TEXT 122
- PL1
 - with TEXT 122
- PLI
 - with TEXT 122
- PLUS
 - count of 173
 - keyword definition 146
 - message CPX11I 155
- PLUS and FORMAT=21 (Ditto Format) 146
- PO
 - DSORG
 - message CPX21I 161
 - message CPX22I 161
- PRINT 83, 142, 181, 183
 - DATA keyword definition 147
 - message CPX05I 151
 - TEXT keyword definition 123
- print only
 - message CPX27I 164
 - message CPX90A 176
- program interrupt 178
- PS
 - DSORG
 - message CPX16I 159
 - message CPX21I 161
 - message CPX22I 161
- PSB 37, 38

Q

- QSAM 7, 31, 36
 - with SYSUT1 97
 - with SYSUT2 97
 - with SYSUT3 129

R

- R - Random
 - message CPX19I 160
 - with KEY 108
 - message CPX07I 152
 - with KEY1 108
 - with Keys 152
 - with SEGMENT 110
- RAMIS II 47
 - example 47
- RAMIS Interface Record Layout 47
- Random 100, 102
- random key 52
- random keys
 - message CPX39A 166
- range
 - with filters 73, 87
- RBA 168-170
 - message CPX36A 165
- READ 160, 178
- RECFM
 - message CPX16I 159
 - message CPX21I 161
 - message CPX22I 161
- relative to one
 - message CPX06I 151
- relative to zero
 - message CPX06I 151
- REP
 - with REPLACE
 - message CPX15I 157
- REPLACE
 - keyword definition 129
- REPORT
 - with TEXT 122
- RESEQ
 - with COPYDIFF 126
 - message CPX15I 157
- return code
 - 0 - zero 176
 - (0 to 16) 15, 175
 - 16 - sixteen 102, 148, 164-166, 176-179
 - 4 - four 176
 - (4-20) 178
 - 8 - eight 95, 172, 176
 - forced by KILLRC 142
 - zero 142
- RKP
 - message CPX16I 159
 - message CPX21I 161
 - message CPX22I 161
- Roll Your Own 50
- ROSCOE 48
 - example 48
- RPG
 - with TEXT 122
- RRDS

message CPX16I 159
message CPX21I 161
message CPX22I 161

S

S0C3
 abend 131
S80A
 abend 31, 178
Sample Batch JCL 25
Sample COBOL Programs 187
Sample Default Messages 9
Sample Foreground CLIST 25
Sample License Information 9
Sample Load Modules Composed of CSECTs 80
Scatter attribute 79
SCRIPT
 with TEXT 122
SCTR 79
SEG 110
SEGMENT 18, 20, 31, 183
 keyword definition 110
 message CPX18I 160
 message CPX19I 160
 message CPX36A 165
 message CPX37A 165
 unusable
 message CPX76I 174
SEGMENT synchronization
 message CPX64I 170
 message CPX65I 170
 message CPX66A 171
-SEL 157
SEQFLD
 with COPYDIFF 126
 message CPX15I 157
sequence
 ascending
 message CPX36A 165
 descending
 message CPX36A 165
 random
 message CPX36A 165
SERENA Consulting 179
SHISAM 36
SKIPUT1
 keyword definition 94
 message CPX09I 154
 message CPX26I 163
SKIPUT2
 keyword definition 94
 message CPX09I 154
 message CPX26I 163
source code
 Natural 33
SPF
 with DIRECTORY 83
 message CPX25I 163

SPIE 178
SQUEEZE
 keyword definition 121
 message CPX25I 163
SQZ 121
SRCH 160, 178
STAMP
 with COPYDIFF 126
 message CPX15I 157
STOPAFT 150, 171
 keyword definition 95
Synchronization After FILTERing - No KEY 105
synchronization KEY
 CPX23I 162
synchronous error 51
 message CPX92A 177
SYS
 with MODE 94
SYSGEN 54
SYSIN 7, 12, 69
SYSPRINT 8, 12, 14
SYSTEMS
 with MODE 13, 24, 26, 94, 166
 message CPX06I 151
Systems Test in Batch Environment 19
Systems Test in Database Environment 20
SYSUT1 7, 168-171, 176-178, 182, 183
 end of file
 message CPX71I 172
 keyword definition 96
 member counts
 message CPX78I 175
 message CPX21I 161
 message CPX27I 164
 record counts
 message CPX75I 173
SYSUT2 7, 168-171, 176-178, 183
 end of file
 message CPX72I 172
 keyword definition 97
 member counts
 message CPX78I 175
 message CPX22I 161
 record counts
 message CPX75I 173
SYSUT3 8, 13, 129, 176, 177
 keyword definition 129
 message CPX16I 159
 record counts
 message CPX75I 173

T

TEMP
 with COPYDIFF 126
 message CPX15I 157
TEXT 182, 183
 end of 172
 keyword definition 122

- message CPX24A 162
- message CPX25I 163
- message CPX39A 166
- with HELP 139

- Text Keyword Properties 112
- TEXT=COBOL 115
- TEXT=COBOL and PRINT=FULL 116
- TSO Command 55
- TUTORIAL 55

U

- underscore 134
- underscored bytes
 - message CPX74I 173
- unexpected changes 6
- Unit Test in Batch Environment 17
- Unit Test in Database Environment 18
- UPPER
 - with CASE 133
 - message CPX08I 153
- USER
 - with DIRECTORY 83
 - message CPX25I 163

V

- VENDOR 54
- VER
 - with LINE 142
- VERS
 - with COPYDIFF 126
 - message CPX15I 157
- VERTICAL
 - with LINE 142
 - message CPX08I 153
- vertical hex 136
- VSAM 7, 31, 36, 176
 - with SYSUT1 97
 - with SYSUT2 97
 - with SYSUT3 129

W

- \$
 - with TEXT 122
- WILDCARD 182
 - keyword definition 98
 - message CPX06I 151

- with TEXT 122
- WILDCARD TEXT 182
- WYLBUR 49
 - example 49

Y

- YES
 - with FRAME 121
 - message CPX25I 163
 - with HALT 11, 71, 138
 - message CPX05I 151
 - message CPX30A 164
 - with INCLUDE 96, 97
 - with MBRHDR 143
 - message CPX05I 151
 - with PASS 126
 - message CPX15I 157
 - with RESEQ 126
 - message CPX15I 157
 - with STAMP 126
 - example 28
 - message CPX15I 157
 - with TEMP 126
 - message CPX15I 157
 - with VERS 126
 - message CPX15I 157
- YESHHMM
 - with VERS 126

Z

- Z - Zoned
 - with FIELD 85
 - with FIELD1 85
 - with FIELD2 86
 - with KEY 108
 - message CPX07I 152
 - with KEY1 108
 - with KEY2 109
 - with MASK 92
 - with MASK1 92
 - with MASK2 92

Numerics

- 80A
 - abend 31, 178



Interactive
System Productivity Facility/
Program Development Facility

SC34-4118-00

Version 2 Release 3

Guide

MVS

First Edition (June 1987)

This edition applies to Version 2 Release 3 of the licensed program Interactive System Productivity Facility/Program Development Facility (ISPF/PDF or PDF) (5665-317), and to all subsequent releases and modifications until otherwise indicated in new editions or Technical Newsletters. It is for use with Version 2 Release 3 of the licensed program Interactive System Productivity Facility (ISPF) (5665-319), MVS/370 Version 1.3.5 or later, MVS/XA Version 2.1.2 or later, and TSO/E Release 3.

Changes are made periodically to the information herein. Therefore, before using this publication, consult the latest IBM System/370 and 4300 Processors Bibliography (GC20-0001) for the editions that are applicable and current.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM licensed program in this document is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent program may be used instead.

Publications are not stocked at the address given below. Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Dept. T46, P. O. Box 60000, Cary, NC 27511. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

ISPF and ISPF/PDF Version 2 Release 3 for MVS Library

- Provides information about ISPF and ISPF/PDF features in the MVS operating environment to assist in evaluating the products.

General
Information
GC34-4116

- Provides information needed to install the products, select installation options, and modify the distributed release. Also provides information to tailor menus, options, and the online tutorial.

Installation and
Customization
SC34-4117

- Describes dialog management data elements and services from programs or command procedures.

Dialog Management
Guide
SC34-4112

- Provides service syntax information plus examples of dialog management services implemented in commonly used programming languages and command procedures such as PL/I, COBOL, FORTRAN, Pascal, APL2, and CLIST.

Dialog Management
Services and
Examples
SC34-4113

- Provides reference and usage information, along with conceptual and functional descriptions of PDF.

PDF Guide
SC34-4118

- Provides reference and usage information with examples for programmers who are developing libraries and who edit or browse data.

PDF Services
SC34-4119

- Provides reference and usage information for programmers who need to create, control, and access controlled libraries.

Library
Management
SC34-4120

- Describes the PDF editor and provides conceptual, usage and reference information for the ISPF/PDF edit line, primary, and macro commands.

Edit and
Edit Macros
SC34-4121

- Provides tutorial type information, samples and exercises to help users learn about and use ISPF and ISPF/PDF.

Primer
SC34-4122

Publication Bill of Forms for ISPF and ISPF/PDF Version 2 Release 3

The preceding publications can be ordered separately or in a Bill of Forms.
The following Bill of Forms numbers are available:

For ISPF, Bill of Forms SBOF-0420-0 includes:

SC34-4112 Dialog Management Guide
SC34-4113 Dialog Management Services and Examples
SC34-4124 ISPF Command Summary Card
SX66-0209 Binder
SX66-0210 Binder cover inserts

For PDF, Bill of Forms SBOF-0419-0 includes:

SC34-4118 PDF Guide
SC34-4119 PDF Services
SC34-4125 PDF Command Summary Card
SX66-0209 Binder
SX66-0211 Binder cover inserts

For PDF Edit and Edit Macros, Bill of Forms SBOF-0361-0 includes:

SC34-4121 Edit and Edit Macros
SC34-4126 Edit and Edit Macros Command Summary Card
SX66-0212 Binder
SX66-0213 Binder cover inserts

ISPF and ISPF/PDF Library for the VM/SP Environment

The books described on the reverse side of this page support ISPF and ISPF/PDF Version 2 Release 3 running in the MVS environment. If you need information about ISPF and ISPF/PDF Version 2 Release 2 running in the VM/SP environment, see:

SC34-4009 Dialog Management Guide
SC34-4010 Dialog Management Services and Examples
SC34-4011 PDF Guide
SC34-4012 PDF Services
SC34-4013 Library Management
SC34-4014 Edit Macros
SC34-4015 Installation and Customization
SC34-4017 Primer
GC34-4036 General Information
SC34-4064 Edit Macros Command Summary Card
SC34-4065 PDF Command Summary Card
SC34-4101 ISPF Command Summary Card

Who Should Use This Book

This book is for application programmers using the ISPF/Program Development Facility.

General
Information
GC34-4116

Installation and
Customization
SC34-4117

About This Book

This book provides information about using PDF, including:

- Introductory information, such as:
 - Invoking and interacting with PDF
 - Selecting PDF options
 - Understanding the display format
 - Entering commands in PDF.
- Library and data set information, including:
 - Descriptions of the programming libraries, data sets, and member lists under PDF
 - Using LMF to control ISPF libraries
 - Using data set passwords, format definitions, mixed mode, and packed data sets
 - Running a sample PDF session.
- Using the PDF options
- Understanding the output listing formats
- Examples of character translation tables
- Lists of abbreviations
- Descriptions of allocation data sets
- Description of the SuperC program.

Dialog Management
Guide
SC34-4112

Dialog Management
Services and
Examples
SC34-4113

PDF Guide
SC34-4118

PDF Services
SC34-4119

Library
Management
SC34-4120

Edit and
Edit Macros
SC34-4121

Primer
SC34-4122

How This Book's Format Differs from the Version 2 Release 2 Edition

This book is a restructured version of *ISPF/PDF for MVS Program Reference*, Version 2 Release 2 (SC34-4024-1). Changes include:

- Library and book overview pages have been added. See pages iii and v. Also, chapter separator pages now provide an overview of the information each chapter contains.
- The color blue is now used to show examples that can be typed on the screen exactly as they appear in the text.
- Information about the new functions provided for Version 2 Release 3 has been added. See “Summary of Changes for ISPF/PDF Version 2 Release 3” on page ix for more information.
- The information formerly found in Chapters 1 and 2 has been combined to form one comprehensive introductory chapter. This caused the information in each of the following chapters, except Chapter 13, to move to the preceding chapter. The information in Chapter 13 did not move because Chapter 12 is completely new, describing the IBM Products primary option.
- Most of the information about the PDF editor has been removed from “Chapter 5, Edit (Option 2)” and placed in *PDF Edit Macros*. This book has been renamed *PDF Edit and Edit Macros*.
- The PDF usage scenario, which was formerly in Appendix A, is now included at the end of Chapter 2. This caused the PDF listing formats to move up from Appendix B to Appendix A and the character translation tables to move up from Appendix C to Appendix B.
- Appendix C now contains the list of abbreviations formerly found in Appendix G.
- Appendix D now contains information about Assembler H and VS COBOL II allocation data sets. The command summary that Appendix D formerly contained has been removed.
- Appendix E now contains reference information about the SuperC (option 3.12), SuperCE (option 3.13), Search-For (option 3.14), and Extended Search-For (option 3.13.S) utilities. The command syntax summary that Appendix E formerly contained has been moved to *PDF Command Summary for MVS*, Version 2 Release 3 (SC34-4125), except for the Edit command syntax summary. That information is now in *PDF Edit and Edit Macros Command Summary*, Version 2 Release 3 (SC34-4126).
- The dialog development model listings, formerly found in Appendix F, have been moved to *PDF Edit and Edit Macros*.
- The index has been expanded and improved.

This Book's Related Publications

The following list contains the titles and publication numbers of books that are referred to in *PDF Guide*:

- MVS/TSO:

GC26-3875 OS/VS2 MVS Data Management Services Guide
GC26-3902 OS/VS2 MVS Utilities
GC28-0683 MVS OS/VS2 Supervisor Services and Macro Instructions
GC28-1154 MVS/XA Supervisor Services and Macro Instructions
SC28-1134 MVS/XA TSO/E TSO Command Language Reference
SC28-1307 TSO/E Command Language Reference

- VM/CMS:

SC19-6209 CMS Command and Macro Reference

- SCRIPT/VS:

SH20-9186 Document Composition Facility: Generalized Markup Language Starter Set User's Guide

- VS COBOL II:

SC26-4049 VS COBOL II Release 2 Application Programming: Debugging Guide

- Cross System Product/Application Development:

GH23-0500 CSP/AD and CSP/AE General Information
GX23-0900 CSP/AD Reference Summary
SH23-0501 CSP/AD User's Guide
SH23-0502 CSP/AD Operation - Development

- Cross System Product/Application Execution:

GH23-0500 CSP/AD and CSP/AE General Information
GV21-5113 Cross System Product Set: Executive Overview
SH23-0503 CSP/AD and CSP/AE System Administration

- Information/Family:

GC34-4045 Introducing the Information/Family
SC34-4046 Planning and Installing the Information/Family
SC34-4052 Using the Information/Family Licensed Programs

- COBOL Structuring Facility:

GC34-4078 COBOL Structuring Facility Licensed Program Specifications
SC34-4079 COBOL Structuring Facility Re-engineering Concepts
SC34-4080 COBOL Structuring Facility User's Guide and Reference

Summary of Changes

Summary of Changes for ISPF/PDF Version 2 Release 3

New Functions for ISPF/PDF Version 2 Release 3

Additional National Language Support (NLS)

- NLS support has been added for Korean.

Note: NLS aids the development of a project by allowing users to work in a language of their choice. This support is available only when the NLS function for this release has been distributed and is installed on your system.

New Customization Features

See *Installation and Customization* for information about the following new customization features:

- PDF Configuration Table
 - A PDF configuration table has been added to help your installation customize PDF. It can be altered to change PDF defaults and to specify installation exits.
- Support for Installation-Written Exits
 - PDF allows installations to satisfy uncommon processing needs by providing for exits at the following points:
 - Data set list and member list dialogs to filter the list content
 - Data set allocation to change the SVC99 parameter list
 - Print utility to provide installation print support
 - Compress request to supply access to alternate methods of compressing data sets
 - Promote at all levels of an LMF controlled library hierarchy
 - Data set name change exit to allow change of data set names.

New Library Access Services

LMACT and LMDEACT

Activates and deactivates, respectively, LMF controlled hierarchies. Similar to option 8.4.

LMCOMP

Compresses partitioned data sets by using either the new compress request exit or IEBCOPY if the exit is not installed from your dialogs. Similar to the compress function provided by options 3.1 and 3.4.

LMCOPY

Copies partitioned data set members or sequential data sets, allowing pack, lock, replace, and automatic truncation options. Similar to the copy function provided by option 3.3.

LMHIER

Creates a table that shows the structure of the controlled library hierarchy that your dialog can examine.

LMMDISP

Includes member selection lists in your dialogs for concatenations of up to four partitioned data sets. Supports all the member list improvements, listed under "Member List Improvements" on page xi, that have been provided by Version 2 Release 3.

LMMOVE

Moves partitioned data set members or sequential data sets, allowing pack, replace, and automatic truncation options. Similar to the move function provided by option 3.3.

LMMSTATS

Sets and stores ISPF statistics for partitioned data set members that have fixed- or variable-length records. Your dialog can delete, set, or have PDF calculate statistics. Similar to the reset statistics function provided by option 3.5, except *all* statistical information can be updated.

LMPRINT

Can print the following to the list data set, with formatting optional:

- A partitioned or sequential data set
- Members of a partitioned data set
- An index listing of a partitioned data set.

Supports data sets with fixed- and variable-length records. Similar to the print function provided by option 3.1.

LMREVIEW

Creates a data set that shows the controls set for a library. Similar to the review controls function provided by option 8.5.

Member List Improvements

- You can specify a pattern to display a partial member list.
- When you display a member list, you can use the following new and enhanced primary commands:

RESET	Blanks out unwanted line commands and messages.
SAVE	Writes the member list to the ISPF list data set or to a sequential data set for recording purposes or as possible input to another dialog.
SELECT	Selects one or more members for processing. A pattern is used to select a subset of the member list.
SORT	Sorts a member list by any of the statistics fields.
LOCATE	Finds a character string according to how a member list is sorted.
- The Data Set List utility supports an extended, nine-character line command area when you use the M line command to display a member list, which allows you to enter TSO CLISTS as line commands.

Edit and Browse Improvements

- Edit and Browse Interface Services
 - Three new services, Edit Interface (EDIF), Browse Interface (BRIF), and Edit Recovery Interface (EDIREC), allow dialogs to provide their own input, output, and edit recovery for the PDF Edit and Browse functions. Using these services, you can create dialogs to edit or browse data from virtually any source. Your dialogs can do pre- and postprocessing on the data. The EDIF service also provides for “hidden” data to be associated with each record and gives indications of which lines were changed, copied, moved, and so forth.
- New Return Code for the EDIT Service and Edit Macro EDIT Command
 - A new EDIT service return code, 14, is issued when you try to edit a data set or PDS member that is already being edited.
- New DISABLED Parameter for Edit DEFINE Command
 - The new DISABLED parameter prevents any built-in command or macro from executing. The command or macro stays disabled until the current Edit session ends and a new one begins.
- New Edit Recovery Table Variables
 - Two new edit recovery table variables, ZEDUSER and ZEUSER, allow you to include any desired data in the Edit Recovery Table by setting the variables to any string. Thereafter, Edit restores the

value during Edit recovery. These variables would be most useful for:

- Dialogs that use the EDREC service (ZEDUSER), or EDIF and EDIREC services (ZEIUSER)
- Edit profiles that include an IMACRO or RMACRO.

New Expiration Date Formats

- Besides YY/MM/DD, the following expiration date formats are now available:
 - YYYY/MM/DD** Allows a four-digit year. The sequence of month, day, and year depends on your national language format.
 - YY.DDD or YYYY.DDD** Allows a Julian format using either a two- or four-digit year.
 - DDDD** Represents the number of days that must pass (0 to 9999).
- A Confirm Purge panel has also been added to help you avoid deleting an unexpired data set by mistake. Previously, unexpired data sets could not be deleted.

Improvements to the Data Set List Utility (Option 3.4)

- You can now use TSO commands and CLISTs as line commands.
- Additional “Z” variables, which contain the data set list information from the display, are available for dialog use. User-written line commands can use this information.
- Two new built-in data set list line commands have been added:

F	Frees unused space.
=	Repeats the last line command entered.

- Four new data set list primary commands have been added:

FIND and RFIND

Finds a string in the Data Set Name field.

SAVE

Writes the data set list to the ISPF list data set or to a sequential data set for recording purposes or as possible input to another dialog.

SHOWCMD

Displays line commands and their arguments before they are invoked, allowing you to change or cancel them.

- Additional keywords have been added to allow the SORT and LOCATE commands to use any field in the data set list.

- The function formerly provided by the VTOC utility (option 3.7) has been incorporated into the Data Set List utility. The VTOC utility has been removed.
- The E (edit data set) line command, and TSO commands and CLISTs are available from the Data Set List utility member lists (M line command).

SuperC, SuperCE, Search-For, and Extended Search-For Utilities

- The SuperC utility (option 3.12) uses the SuperC program to compare two or more data sets. SuperC creates output listings that show the differences between the data sets being compared.
- The SuperCE utility (option 3.13) is an extended version of option 3.12 for the more sophisticated user. With this utility, you can use various options to tailor the comparison to fit your data. Once you have done that, you can save the options as a profile for use with the simpler SuperC utility (option 3.12).
- The Search-For utility (option 3.14) allows you to search multiple data sets for strings of data.
- The Extended Search-For utility (option 3.13.S) is an extended version of option 3.14. Like SuperCE, it provides options that give the sophisticated user more flexibility.

Assembler H and VS COBOL II Support

- Assembler H, the VS COBOL II compiler, and VS COBOL II interactive debug have been added to the selection menus for Foreground (option 4) and Batch (option 5). The existing system assembler has been retained and designated Assembler XF.

Enhancements to the Dialog Test Variables Display

- The Dialog Test Variables (option 7.3) display now uses two new pool indicators:

V	Function pool; the variable was defined with the VDEFINE service.
I	Function pool; the variable is implicit.
- Function pool variables you insert on the display by using the I line command or by typing over a variable name are indicated as implicit variables. Also, you can now use the D line command to delete modifiable shared and profile pool variables.
- A new APPLICATION display field has been added to show the name of the application to which the variables belong.

Library Management Facility (LMF) Improvements

- LMF Lock Function Added to the Move/Copy Utility
- Two new Move/Copy utility options allow you to lock a data set or member when you copy it into your private library. These options are:
 - L** Copy and lock member(s).
 - LP** Copy, lock, and print member(s).
- New LMF Functions When a Data Set is Full
 - The library administrator can now specify one of three actions for LMF to take if the control data set, the activity logs, or any controlled libraries run out of space. The action can be specified by using the LMF Controls utility (option 8.1) or the new LMF Data Set Full Action utility (option 8.7). The three actions are:
 - FAIL** LMF tells the user that the requested function failed.
 - NOTIFY** Same as FAIL, except LMF also notifies the library administrator.
 - EXIT** LMF calls an installation-supplied exit routine.
- Activity Log Additions and Improvements
 - A new activity log has been added that, if preallocated, records all library administrator functions. Also, the user's ISPF activity log now records member promotion.

The new LMF Activity Log utility (option 8.6) lets the library administrator reduce the size of existing activity logs. Other users can use this utility to browse any activity log to which they have access authority.
- LMF Utility to Manage Member Controls
 - Functions have been added to the LMF Controls utility (option 8.1) to let the library administrator initialize, clear, and verify member controls. It is now possible for the library administrator to create controls for existing libraries without having to create a new library and promote all the members of the existing library into it.
- Deleting LMF Controlled Members or Versions
 - The LMF Controls utility (option 8.1) has been changed to delete a member when you delete its controls. Previously, when you used this option to delete a member's controls, the member still existed. You can also use this option to delete just a specific version within the hierarchy.

- Multiple LMF Started Task Names
 - LMF now supports multiple control tasks coupled to individual control files. This allows your installation to isolate library controls for different groups to provide more specific security and accounting.
- LMF Review Control Status Listing Additions
 - Listings created with the LMF Review utility (option 8.5) now contain the following additional information:
 - Promotion exits
 - Actions taken when the following data sets are full:
 - Activity log
 - Library
 - Library administrator activity log
 - Control data set.
 - LMF user group names.
- New LOCK Field on Edit - Entry Panel
 - The LOCK field lets you specify whether to lock the member you will edit. If you specify YES and the member is part of a controlled LMF library, no one else can change the member until you promote it. If you specify NO, the member is not locked, anyone else can change it, and you cannot promote your changes. If you specify NEVER, the member is not locked and never will be unless you change this value.

IBM Products Option

- A new primary option called IBM Products (option 9) has been added to the ISPF/PDF Primary Option Menu. This option lets you invoke the following products if they are available and installed:
 - Cross System Product/Application Development (CSP/AD)
 - Cross System Product/Application Execution (CSP/AE)
 - Information/System (INFO/SYS)
 - COBOL Structuring Facility foreground dialog (COBOL/SF-F)
 - COBOL Structuring Facility background dialog (COBOL/SF-B).

Contents

Chapter 1. Introduction to PDF	1
PDF Uses	1
Browse, Edit, Edit Macros, and Models	2
Dialog Services	2
Utilities	2
Dialog Test, Foreground, and Batch	3
TSO Commands and CLISTs	3
Library Management Facility	3
Other IBM Program Development Products	3
Primer and Online Tutorial	4
Interacting with PDF	4
Invoking PDF	4
Option Selection	5
ISPF/PDF Primary Options	5
Understanding PDF Panels	7
Panel Format	7
Panel Types	8
Menus	9
Data Entry Panels	9
Scrollable Data Displays	9
Entering Commands in PDF	10
Types of Commands	10
TSO Commands and CLISTs	11
ISPF Commands	11
ISPF Commands Assigned to PF Keys	11
PDF Primary and Line Commands	14
Methods of Entering Commands	14
TSO Command Processor Panel (Option 6)	14
COMMAND and OPTION Fields	14
TSO Commands and CLISTs	15
ISPF Commands	15
PDF Primary Commands	15
ISPF/PDF Command Notation	15
Stacking Commands	16
Dual Command Processing	16
Line Command Fields	16
TSO Commands and CLISTs	17
PDF Line Commands	18
Program Access (PA) Keys	18
Program Function (PF) Keys	19
Default ISPF Command Assignments	19
Using the Default PF Key Settings	20
Getting Help	20
Splitting and Swapping Display Screens	21
Using the SPLIT Command	21
Repositioning a Split Screen	21
Split Screen Limitations	21

Using the SWAP Command	22
Ending PDF or a PDF Function	22
Using the END Command	23
Using the RETURN Command	24
Using the Exit Option (X)	25
Repeating a FIND or CHANGE Command	25
Using the RFIND Command	25
Using the RCHANGE Command	25
Scrolling Within a Data Display	25
Using the Scroll Commands	26
Typing Over Scroll and Multiple-Choice Parameters	26
Retrieving Commands	27
How ISPF Stores Commands	28
How ISPF Retrieves Commands	28
Remembering Information with User Profiles	28
 Chapter 2. Libraries and Data Sets	29
Naming ISPF Libraries and Data Sets	29
ISPF Library Names	29
Other Partitioned or Sequential Data Set Names	31
Volume Serials	32
Using LMF to Control ISPF Libraries	32
Library Concatenation	34
Concatenation During Editing	35
Concatenation During Language Processing	35
Library Member Change and Promotion	36
Locking Members While Editing	36
Conditions When Locking is Ignored	37
Conditions When Locking Causes Errors	37
Using Member Selection Lists	37
Displaying Member Lists	38
Ending Member Lists	39
ISPF Member Statistics	39
Load Module Library Member Statistics	41
Member Selection List Commands	42
Locating a Data String	43
Removing Unwanted Line Commands and Messages	44
Writing a Member List to a Sequential Data Set	44
Selecting a Member	45
SELECT Primary Command	45
S Line Command	45
Sorting a Member List	46
Library and Data Set List Utility Line Commands	47
Updating a Member List	49
Using Data Set Passwords	49
Using Format Definitions	50
Using Mixed Mode	50
Using Packed Data Sets	51
Using List and Log Data Sets	52
List Data Set	53
Log Data Set	53
Printing List/Log Data Sets	53
Foreground and Batch Output Listings	54
Other Temporary Data Sets	54

Job Statement Information	55
Running a Sample PDF Session	56
Chapter 3. ISPF Parms (Option 0)	63
Terminal (Option 0.1)	64
Log/List (Option 0.2)	67
Log and List Data Set Default Options	68
Changing Primary and Secondary Page Values	69
PF Keys (Option 0.3)	70
Terminals with 12 PF Keys	70
Terminals with 24 PF Keys	70
Assigning ISPF/PDF Commands	70
Assigning Labels	72
Display (Option 0.4)	74
List (Option 0.5)	76
Printing Panels with Long Lines	77
Graphic (Option 0.6)	78
Environ (Option 0.7)	81
Using the DUMP Parameter	84
Chapter 4. Browse (Option 1)	85
Browsing a Data Set	85
Browse Data Display	86
Ending Browse	87
Browse Primary Commands	87
BROWSE - Browsing Recursively	87
COLUMNS - Identifying Columns	88
DISPLAY - Controlling the Display	89
FIND - Finding Character Strings	90
Specifying Find Strings	90
Omitting String Delimiters	91
Using String Delimiters	91
Starting Point, Direction, and Extent of Search	91
Conditions for Character String Matches	92
Use of Text Strings	92
Use of Character Strings	93
Use of Picture Strings	93
Column Limitations	94
Using RFIND	94
HEX - Displaying Data in Hexadecimal Format	95
LOCATE - Locating Lines	96
RESET - Removing the Column-Identification Line	97
Chapter 5. Edit (Option 2)	99
Editing a Data Set	99
Chapter 6. Utilities (Option 3)	101
Library Utility (Option 3.1)	102
Blank - Display Member List	102
B - Browse Member	103
C - Compress Data Set	103
P - Print Member	103
X - Print Index Listing	103
R - Rename Member	103

L - Print Entire Data Set	104
D - Delete Member	104
I - Data Set Information	104
S - Data Set Information (Short)	105
Data Set Utility (Option 3.2)	106
A - Allocate New Data Set	106
C - Catalog Data Set	111
R - Rename Entire Data Set	111
U - Uncatalog Data Set	112
D - Delete Entire Data Set	112
S - Data Set Information (Short)	114
Blank - Data Set Information	114
Move/Copy Utility (Option 3.3)	115
C and CP - Copying Data Sets	116
C - Copy Data Set or Member(s)	116
CP - Copy and Print	116
M and MP - Moving Data Sets	118
M - Move Data Set or Member(s)	118
MP - Move and Print	118
L and LP - Copying and Locking Data Sets	119
L - Copy and Lock Member(s)	119
LP - Copy, Lock, and Print	119
P and PP - Promoting Data Sets	120
P - Promote Data Set or Member(s)	120
PP - Promote and Print	120
Using the Move/Copy Utility without Load Modules	122
Using the Move/Copy Utility with Load Modules	122
Moving, Copying, or Promoting Alias Entries	123
Data Set List Utility (Option 3.4)	124
Blank - Display Data Set List	127
Line Commands	131
TSO Commands and CLISTS	132
Using the Slash (/) Symbol	132
TSO Command/CLIST Variables	134
P - Print Data Set List	135
V - Display VTOC Information	136
PV - Print VTOC Information	137
Data Set List Utility Line Commands	138
B - Browse Data Set	138
E - Edit Data Set	138
D - Delete Data Set	138
R - Rename Data Set	139
I - Data Set Information	139
S - Information (Short)	139
C - Catalog Data Set	139
U - Uncatalog Data Set	139
P - Print Data Set	139
X - Print Index Listing	139
M - Display Member List	139
Z - Compress Data Set	141
F - Free Unused Space	141
= - Repeat Last Command	141
Data Set List Utility Primary Commands	142
CONFIRM Command	142

FIND and RFIND Commands	142
LOCATE Command	143
SAVE Command	143
SHOWCMD Command	144
SORT Command	146
Reset ISPF Statistics Utility (Option 3.5)	147
R - Reset (Create/Update) ISPF Statistics	148
D - Delete ISPF Statistics	149
Results of Resetting Statistics	149
Hardcopy Utility (Option 3.6)	150
Generating and Submitting JCL	151
Additional Batch Printing Information	153
Using the TSO/E Information Center Facility	153
Removal of List VTOC Utility (Option 3.7)	155
Outlist Utility (Option 3.8)	156
L - List Job Names/IDs Via the TSO STATUS Command	157
D - Delete Job Output from SYSOUT Hold Queue	157
P - Print Job Output and Delete from SYSOUT Hold Queue	157
R - Requeue Job Output to a New Output Class	158
Blank - Display Job Output	158
Command Table Utility (Option 3.9)	159
Scrolling a Command Table	162
Saving a Command Table	162
Cancelling a Command Table	162
Command Table Line Commands	162
D - Deleting Lines	162
I - Inserting Lines	162
R - Repeating Lines	162
Convert Menus/Messages (Option 3.10)	163
1 - Convert Old Format Menus to New Format Panels	164
2 - Convert Old Format Messages to New Format Messages	165
Format Specification Utility (Option 3.11)	166
A - Add a New Format	167
C - Copy Formats	168
D - Delete a Format	169
U - Update a Format	169
L or BLANK - Display Format List	169
Format Selection List Commands	170
Deleting a Format	170
Locating Format Names	170
Renaming a Format	171
Sorting Format Names	171
Updating or Selecting a Format	171
SuperC Utility (Option 3.12)	172
SuperC Member Lists	176
Submitting a SuperC Job in Batch Mode	177
Blank - Generate Output Listing to SYSOUT CLASS	179
1 - Generate Output Listing in DATA SET NAME	179
Specifying Existing Data Sets	179
Specifying Nonexistent Data Sets	180
2 - Generate Output Listing Using Completed //OUTDD DD	180
Using the NOLIST Listing Type in Batch Mode	181
SuperCE Utility (Option 3.13)	182
Blank - Compare Data Sets	186

B - Submit Batch Data Set Compare	187
Printing a SuperCE Listing in Batch Mode	188
S - Extended Search-For Compare Utility	188
Blank - Search-For Strings	191
P - Select Search-For Process Options	192
E - Edit Search-For Statements Data Set	193
P - Select SuperCE Process Options	194
E - Edit SuperCE Statements Data Set	194
A - Activate/Create Profiles	195
A - Activate	197
C - Create	197
Search-For Utility (Option 3.14)	199
Entering Multiple Search Strings	200
Search-For Strings and Keywords	201
Entering Search Strings	201
Using Keywords	201
Search-For Member Lists	202
 Chapter 7. Foreground (Option 4)	205
Foreground Processing Sequence	205
Expanding Packed Data	211
Restrictions on Member Expansion and Member Parts Lists	213
Member Expansion Return Codes	215
Trigger Statement Errors	215
Input Data Sets	215
List Data Sets	216
Password Protection	217
Object Data Sets	217
Foreground - TSO/E Information Center Facility	218
Assembler H (Option 4.1)	219
Assembler XF (Option 4.1A)	220
VS COBOL II Compiler (Option 4.2)	221
OS/VS COBOL Compiler (Option 4.2A)	223
VS FORTRAN Compiler (Option 4.3)	225
PL/I Checkout Compiler (Option 4.4)	226
Interpretable Text Data Sets	227
PL/I Optimizing Compiler (Option 4.5)	228
Pascal/VS Compiler (Option 4.6)	229
Linkage Editor (Option 4.7)	231
Linkage Editor Concatenation Sequence	232
SCRIPT/VS (Option 4.9)	233
SCRIPT/VS Processing Sequence	233
Selecting a Formatting Style	237
Changing Style Options	238
Using SCRIPT/VS with the TSO/E Information Center Facility	239
VS COBOL II Interactive Debug (Option 4.10)	240
COBOL Interactive Debug (Option 4.10A)	241
COBOL Interactive Debug Processing Sequence	241
Symbolic Debug Data Sets	242
Print Output Data Sets	243
FORTRAN Interactive Debug (Option 4.11)	244
Member Parts List (Option 4.12)	245
Member Not Found	249

Chapter 8. Batch (Option 5)	251
Batch Processing Sequence	251
JCL Generation - Compilers	256
JCL Generation - Assemblers and Linkage Editor	257
Assembler H (Option 5.1)	258
Assembler XF (Option 5.1A)	260
VS COBOL II Compiler (Option 5.2)	262
OS/VS COBOL Compiler (Option 5.2A)	264
VS FORTRAN Compiler (Option 5.3)	265
PL/I Checkout Compiler (Option 5.4)	267
PL/I Optimizing Compiler (Option 5.5)	268
Pascal/VS Compiler (Option 5.6)	269
Linkage Editor (Option 5.7)	270
VS COBOL II Interactive Debug (Option 5.10)	272
Member Parts List (Option 5.12)	273
 Chapter 9. Command (Option 6)	 275
Entering TSO Commands and CLISTs	276
Rules for Entering TSO Commands	276
Rules for Entering CLISTs	277
Using the Session Manager	277
 Chapter 10. Dialog Test (Option 7)	 279
Environment	280
Variable Usage	281
Severe Error Handling	282
Commands	282
Ending the Current Option Without Saving Changes	284
Saving Changes	284
Finding a Character String	284
Displaying Breakpoint Qualification Data	285
Restoring the Format of the Breakpoints Panel	285
Dialog Test Line Commands	285
D - Deleting Lines	285
I - Inserting Lines	285
R - Repeating Lines	286
Functions (Option 7.1)	287
Panels (Option 7.2)	291
Variables (Option 7.3)	293
Variables Commands	294
Manipulating Variables	294
Creating New Variables	295
Deleting Variables	295
Variables Usage Notes	295
Tables (Option 7.4)	298
1 - Display Row	300
Display Row Commands	301
2 - Delete Row	301
3 - Modify Row	301
Modify Row Commands	303
4 - Add Row	304
Add Row Commands	305
5 - Display Structure	306

Display Structure Command	306
6 - Display Status	307
Table Not Open	307
Table Open	308
Log (Option 7.5)	310
ISPF Log Not Available	310
Trace Output in ISPF Log	310
Trace Header Entries	310
Function Trace Entries	311
Variable Trace Entries	311
Dialog Services (Option 7.6)	313
Traces (Option 7.7)	315
1 - Function Traces	315
Function Traces Commands	317
2 - Variable Traces	317
Variable Traces Commands	318
Breakpoints (Option 7.8)	319
Specifying Breakpoints	319
Breakpoints Commands	320
Qualification Parameter Values	320
Finding a Breakpoint	323
Tutorial (Option 7.T)	326
Exit (Option 7.X)	326
 Chapter 11. LM Utilities (Option 8)	327
 Chapter 12. IBM Products (Option 9)	329
CSP/AD (Option 9.1)	330
CSP/AE (Option 9.2)	331
INFO/SYS (Option 9.3)	332
COBOL/SF-F (Option 9.4)	333
COBOL/SF-B (Option 9.5)	334
 Chapter 13. Tutorial (Option T)	335
Selecting the Tutorial Option (T)	335
Selecting a Tutorial Topic	335
Ending the Tutorial	337
Using the HELP Command	338
 Appendix A. Listing Formats	339
Source and Index Listings	339
Source Listings	340
Index Listings	341
Index Listings for Source Libraries	341
Index Listings for Load Libraries	342
ISPF Log Listings	343
Member List Listings	344
Member List Listings for Source Libraries	344
Member List Listings for Load Libraries	345
Formats for Member List Listings	346
Data Set List Listings	347
SuperC Listings	348
SuperC Listing Title Lines	349
Listing Output Section	352

Listing Prefix Codes	352
Type of Difference Codes	353
Member Summary Listing	355
Summary and Statistics/PDS Overall Totals	355
Bottom-of-Listing Information	356
Side-By-Side Listings	357
Effect of FILE Compare Type on SuperC Listings	358
FILE Comparison of a Sequential Data Set or Membered PDS ..	358
FILE Comparison of a Complete PDS	359
Search-For Listings	360
 Appendix B. Character Translations for APL and TEXT	
Keyboards	363
 Appendix C. List of Abbreviations	365
Command Name Abbreviations	365
Field Value Abbreviations	365
Keyword/Operand Abbreviations	366
Scroll Amount Abbreviations	366
Programming Language Abbreviations	366
 Appendix D. Allocation Data Sets	367
SYSIN Data Set	367
SYSLIB Data Set	367
SYSPRINT Data Set	368
SYSTEM Data Set	368
SYSLIN Data Set	368
SYSPUNCH Data Set	369
SYSUT1 Data Set	369
SYSUT2 - SYSUT5 Data Sets	369
 Appendix E. SuperC Reference	371
Utility Differences	371
Program Description	372
Comparison Result Differences	374
Applications	375
Return Codes	377
Process Options	378
Input Process Control Options	379
"Don't Process" Control Options	380
Output Process Control Options	381
Update Data Set Control Options	382
APNDUPD - Append Update Data Set	383
UPDCMS8 - Update CMS Sequenced 8	383
UPDCNTL - Update Control	384
UPDCNTL Data Set Using LINE Compare	384
UPDCNTL Data Set Using WORD Compare	385
UPDCNTL Data Set Using BYTE Compare	386
UPDMVS8 - Update MVS Sequenced 8	387
UPDPDEL - Prefixed Delta Lines	388
UPDSEQ0 - Update Sequenced 0	389
UPDSUMO - Update Summary Only	389
UPDSUMO Data Set Using LINE Compare	390
UPDSUMO Data Set Using WORD Compare	391

UPDSUMO Data Set Using BYTE Compare	392
Process Statements	393
Process Statement Notation Conventions	393
CMPBOFS - Compare Byte Offsets	395
Compare Format	395
Parameters	395
Example	395
CMPCOLM - Compare Columns	396
Compare and Search Format	396
Parameters	396
Example	396
CMPCOLMN/CMPCOLMO - Compare Column New and Old	397
Compare Format	397
Parameters	397
Example	398
CMPLINE - Compare Line	399
Compare and Search Format	399
Parameters	399
Examples	400
Comment Lines	401
Compare and Search Format	401
DPLINE/DPLINEC - Don't Process This Line	402
Compare and Search Format	402
Parameters	402
Examples	403
LNCT - Line Count	404
Compare and Search Format	404
Parameters	404
Example	404
LSTCOLM - List Columns	405
Compare and Search Format	405
Parameters	405
Example	405
NCHGT/OCHGT - New and Old Change Text	406
Compare Format	406
Search Format	406
Parameters	406
Examples	407
NTITLE/OTITLE - Alternate Title for Input Data Sets	408
Compare Format	408
Search Format	408
Parameter	408
Example	408
SELECT - Select PDS Members	409
Compare Format	409
Search Format	409
Compare Parameters	409
Search Parameter	409
Examples	409
Compare Example	409
Search Example	410
SLIST - Include Statements in Listing	411
Compare and Search Format	411
Parameters	411

Example	411
SRCHFOR/SRCHFORC - Search-For String	412
Search Format	412
Parameters	412
Examples	412
Index	413

Figures

1.	ISPF/PDF Primary Option Menu	5
2.	Panel Format	7
3.	Command Types and Entry Methods	10
4.	ISPF Commands Assigned to PF Keys	12
5.	Other ISPF Commands	13
6.	Line Command Field Characteristics	16
7.	Default PF Key Settings	20
8.	Split Screen Example	21
9.	Specify Disposition of Log and List Data Sets Panel	23
10.	Hierarchy of ISPF Libraries	33
11.	Member Selection List Differences	38
12.	Member List Display	40
13.	Load Module Library Display	41
14.	SORT Fields for Source Libraries	46
15.	SORT Fields for Load Libraries	46
16.	Library Utility - Before and After Print, Rename, and Delete	48
17.	ISPF Parameter Options Panel	63
18.	Terminal Characteristics Panel	64
19.	Log and List Defaults Panel	67
20.	PF Key Definitions and Labels Panel (12 PF Keys)	70
21.	PF Key Definitions and Labels Panels (24 PF Keys)	71
22.	Sample Screen with PF Key Definition Lines	73
23.	Display Characteristics Panel	74
24.	List Data Set Characteristics Panel	76
25.	Print Graphics Parameters Panel	78
26.	Screen Containing Graphics to be Printed	79
27.	Example of Using Aspect Ratio Parameter 0	79
28.	Example of Using Aspect Ratio Parameter 1	80
29.	ISPF ENVIRON Command Settings Panel	81
30.	Browse - Entry Panel	85
31.	Browse - Data Display	86
32.	Browse - Column-Identification Line	89
33.	Browse Hexadecimal Display - Vertical	96
34.	Browse Hexadecimal Display - Data	96
35.	Edit - Entry Panel	99
36.	Utility Selection Menu	101
37.	Library Utility Panel	102
38.	Library Utility - Data Set Information	104
39.	Library Utility - Data Set Information (Short)	105
40.	Data Set Utility Panel	106
41.	Allocate New Data Set Panel - Two-digit Year Support	107
42.	Allocate New Data Set Panel - Four-digit Year Support	110
43.	Rename Data Set Panel	112
44.	Confirm Delete Panel	112
45.	Confirm Purge Panel	113
46.	Move/Copy Utility Panel	115
47.	Move/Copy Utility - "To" Panel for Copying	116

48.	Move/Copy Utility - "To" Panel for Moving	118
49.	Move/Copy Utility - "Target" Panel for Promoting	121
50.	Data Set List Utility Panel	124
51.	Sequence of Data Set List Display Views	126
52.	Data Set List - VOLUME View	128
53.	Data Set List - SPACE View	128
54.	Data Set List - ATTRIB View	129
55.	Data Set List - TOTAL View	129
56.	Data Set List Utility Line Commands and Messages	130
57.	TSO Command/CLIST Variables (Output)	134
58.	TSO Command/CLIST Variables (Input)	135
59.	VTOC Information Display	136
60.	Member List Display - Expanded Line Command Field	140
61.	Load Module Library Display - Expanded Line Command Field	141
62.	Data Set List Utility - SHOWCMD Panel	145
63.	SORT Fields for Source Libraries	146
64.	Reset ISPF Statistics Panel	147
65.	Reset Statistics Field Combinations and Results	149
66.	Hardcopy Utility Panel - Before JCL Generation	150
67.	Hardcopy Utility Panel - After JCL Generation	152
68.	Hardcopy Utility Panel - With the TSO/E Information Center Facility Installed	154
69.	Outlist Utility Panel	156
70.	Command Table Utility Panel	159
71.	Command Table Editing Panel	160
72.	Convert Menus/Messages Panel	163
73.	Copy Panel for Specifying New Format Data Set	164
74.	Format Specification Panel	166
75.	Format Definition Panel	167
76.	Copy Format Selection List Panel	169
77.	Format Selection List Panel	170
78.	SuperC Utility Panel	172
79.	SuperC Utility - "Old" Data Set Panel	173
80.	SuperC Member List Panel	177
81.	SuperC Utility - Submit Batch Jobs Panel	178
82.	SuperC Utility - Submit Batch Jobs Panel using NOLIST	181
83.	SuperCE Utility Panel	182
84.	SuperCE - Concatenation Interactive Entry Panel	187
85.	SuperCE - Concatenation Batch Entry Panel	188
86.	Extended Search-For Utility Panel	189
87.	Extended Search-For - Concatenation Data Set Entry Panel	192
88.	Search-For Process Statements Panel	193
89.	SuperC Process Statements Panel	195
90.	SuperCE Profile Browse Display	196
91.	SuperCE - Profile Manager Panel	196
92.	Search-For Utility Panel	199
93.	Multiple Search Strings Panel	201
94.	Search Member List Panel	203
95.	Foreground Selection Panel	205
96.	Foreground Print Options Panel	210
97.	Expansion Triggers and Syntax	212
98.	Foreground Print Options Panel With TSO/E Information Center Facility	218
99.	Foreground Assembler H Panel	219
100.	Foreground Assembler H Options	219

101.	Foreground Assembler XF Panel	220
102.	Foreground Assembler XF Options	220
103.	Foreground VS COBOL II Compile Panel	221
104.	Foreground VS COBOL II Compiler Options	222
105.	Foreground OS/VS COBOL Compile Panel	223
106.	Foreground OS/VS COBOL Compiler Options	224
107.	Foreground VS FORTRAN Compile Panel	225
108.	Foreground VS FORTRAN Compiler Options	225
109.	Foreground PL/I Checkout Compile Panel	226
110.	Foreground PL/I Checkout Compiler Options	227
111.	Foreground PL/I Optimizing Compile Panel	228
112.	Foreground PL/I Optimizing Compiler Options	228
113.	Foreground Pascal/VS Compile Panel	229
114.	Foreground Pascal/VS Compiler Options	229
115.	Foreground Linkage Edit Panel	231
116.	Foreground Linkage Editor Options	231
117.	SCRIPT/VS Processor Panel	233
118.	Foreground Print Options for Style Panel	236
119.	Select SCRIPT/VS Formatting Style Panel	237
120.	SCRIPT/VS Options for Style Panel: Draft	238
121.	Foreground Print Options for Style Panel With the TSO/E Information Center Facility	239
122.	Foreground VS COBOL II Interactive Debug Panel	240
123.	COBOL Interactive Debug Panel	241
124.	FORTRAN Interactive Debug Panel	244
125.	Foreground FORTRAN Interactive Debug Options	244
126.	Foreground Member Parts List Panel	245
127.	Member Parts List Display	247
128.	Foreground Member Parts List Record Formats	248
129.	Batch Selection Panel	251
130.	Batch Selection Panel With JCL Generated	255
131.	ISRSCAN and ISRLEMX Return Codes	257
132.	Batch Assembler H Panel	258
133.	Batch Assembler H Options	259
134.	Batch Assembler XF Panel	260
135.	Batch Assembler XF Options	261
136.	Batch VS COBOL II Compile Panel	262
137.	Batch VS COBOL II Compiler Options	263
138.	Batch OS/VS COBOL Compile Panel	264
139.	Batch OS/VS COBOL Compiler Options	264
140.	Batch VS FORTRAN Compile Panel	265
141.	Batch VS FORTRAN Compiler Options	266
142.	Batch PL/I Checkout Compile Panel	267
143.	Batch PL/I Checkout Compiler Options	267
144.	Batch PL/I Optimizing Compile Panel	268
145.	Batch PL/I Optimizing Compiler Options	268
146.	Batch Pascal/VS Compile Panel	269
147.	Batch Pascal/VS Compiler Options	269
148.	Batch Linkage Edit Panel	270
149.	Batch Linkage Editor Options	271
150.	VS COBOL II Interactive Debug Panel	272
151.	Batch Member Parts List Panel	273
152.	TSO Command Processor Panel	275
153.	Dialog Test Primary Option Menu	280
	Invoke Dialog Function/Selection Menu	287

155.	Display Panel	291
156.	Message Display Panel	292
157.	Variables Panel	293
158.	Tables Panel	298
159.	Display Row Panel	300
160.	Confirm Table Row Delete Panel	301
161.	Modify Row Panel	302
162.	Add Row Panel	304
163.	Table Structure Panel	306
164.	Status for Table Panel with Table Not Open	307
165.	Status for Table Panel with Table Open	308
166.	Invoke Dialog Service Panel	313
167.	Special Display Panel	314
168.	Traces Panel	315
169.	Function Traces Panel	316
170.	Variable Traces Panel	317
171.	Breakpoints Panel	319
172.	Breakpoints Panel with Qualification Parameter Values	321
173.	Breakpoint Primary Option Menu	323
174.	Dialog Test Tutorial - First Panel	326
175.	Library Management Utilities Panel	327
176.	Additional IBM Program Development Products Panel	329
177.	Cross System Product/Application Development Panel	330
178.	Cross System Product/Application Execution Panel	331
179.	Information/System Panel	332
180.	COBOL Structuring Facility Panel - Foreground Dialog	333
181.	COBOL Structuring Facility Panel - Background Dialog	334
182.	ISPF Tutorial and Introduction Panels	336
183.	Table of Contents Panel	337
184.	Help Tutorial Panel	338
185.	Sample Source Listing	340
186.	Sample Index Listing - Source Library	341
187.	Sample Index Listing - Load Library	342
188.	Sample ISPF Log Listing	343
189.	Sample Member List Listing for a Source Library	344
190.	Sample Member List Listing for a Load Library	345
191.	Format of Source Library Member List Listing	346
192.	Format of Source Library Member List Listing	346
193.	Sample Data Set List Listing	347
194.	Format of Data Set List Listing	347
195.	Split Screen Showing Source Data for SuperC Listings	349
196.	DELTA Listing of a LINE Comparison of Two PDSs	350
197.	LONG Listing of a WORD Comparison of Two PDSs	351
198.	WORD Comparison of Two PDSs Using Change Bar to Show Differences	354
199.	Side-By-Side LINE Comparison Using the NARROW Process Option	357
200.	FILE Comparison of a Membered PDS	358
201.	FILE Comparison of a Complete PDS	359
202.	Search-For Listing	360
203.	Internal Character Representations for APL Keyboards	363
204.	Internal Character Representations for TEXT Keyboards	364
205.	Input Process Control Options	379
206.	"Don't Process" Control Options	380
207.	Output Process Control Options	381

208.	UPDCMS8 Update Data Set	383
209.	UPDCNTL Data Set Using LINE Compare	384
210.	UPDCNTL Format Using LINE Compare	384
211.	UPDCNTL Listing Using WORD Compare	385
212.	UPDCNTL Format Using WORD Compare	385
213.	UPDCNTL Listing Using BYTE Compare	386
214.	UPDCNTL Format Using BYTE Compare	387
215.	UPDMVS8 Update Data Set Listing	387
216.	UPDPDEL Update Data Set Listing	388
217.	UPDSEQ0 Update Data Set	389
218.	UPDSUMO Data Set Using LINE Compare	390
219.	UPDSUMO Format Using LINE Compare	390
220.	UPDSUMO Data Set Using WORD Compare	391
221.	UPDSUMO Format Using WORD Compare	391
222.	UPDSUMO Data Set Using BYTE Compare	392
223.	UPDSUMO Format Using BYTE Compare	392

CHAPTER 1. INTRODUCTION

This chapter introduces you to the Program Development Facility (PDF) and its capabilities by explaining:

- **How to begin and end a PDF session**
- **PDF primary options**
- **PDF panel formats and types**
- **How to enter ISPF commands and use the default PF key settings**
- **How PDF “remembers” information by storing it in your user profile.**

The chapter also explains, in general terms, how you can use PDF’s facilities to perform your programming tasks.

Chapter 1. Introduction to PDF

PDF stands for Program Development Facility. The prefix ISPF/, as in ISPF/PDF, shows that the Interactive System Productivity Facility is PDF's prerequisite licensed program. ISPF must be installed either before or during the installation of PDF.

As its name implies, PDF's primary purpose is to help programmers develop interactive applications, which we call *dialogs*. Dialogs are called "interactive" applications because PDF uses them to communicate with terminal users through a series of panels while the users do application development tasks. PDF panels:

- Provide access to ISPF/PDF functions (menus)
- Request information (data entry panels)
- Provide information (scrollable data displays).

The following sections describe PDF users and the functions provided by the ISPF/PDF licensed program.

PDF Uses

Here are some of the ways to use PDF:

- Managers use PDF Edit, SCRIPT/VS, and the Hardcopy utility or another print utility provided by their installation to prepare memos
- Data processing administrators and system programmers use PDF to:
 - Monitor and control program libraries
 - Communicate with MVS through TSO commands or CLISTs.
- Programmers use PDF to develop batch, interactive, or any other type of program and its documentation
- Terminal users invoke dialogs that use ISPF and PDF dialog services to do the work of the application.

To help you, the programmer, develop ISPF/PDF dialogs, PDF provides Edit, Browse, utility, compile, program library control, and other facilities.

Browse, Edit, Edit Macros, and Models

The Edit and Browse functions let you look at the contents of a dialog on the terminal screen. While editing a dialog, you can change it by adding or deleting lines, typing over the existing code, or copying lines from another dialog to the one being edited.

To enhance the existing Edit functions, you can write edit macros. Edit macros let you combine several often-used functions so that you do not have to invoke each function separately. You can write initial edit macros that are automatically run when the Edit option is invoked. Other uses for edit macros include:

- Overriding PDF Edit commands
- Invoking ISPF and PDF dialog services
- Accessing cursor position and data location.

Also, PDF provides online models that you can insert into the dialog. A model is an example of a service call, panel format, table format, or message that contains the proper syntax and all the available parameters for the programming language being used. Since these models are online, they can be called directly into the member being edited.

See *Edit and Edit Macros* for complete information about using the PDF Edit option and writing edit macros.

Dialog Services

PDF provides edit, browse, and library access services that can be combined in a dialog with any of the ISPF services. The library access services carry out functions involving members of a programming library. These functions include adding, finding, and deleting members, and displaying member lists.

A separate model of each service call is included for each programming language ISPF/PDF supports: CLIST, COBOL, EXEC, FORTRAN, PL/I, and Pascal. See *PDF Services* for complete information about the PDF dialog services.

Utilities

PDF provides a wide range of utilities. For example, the utilities let you:

- Display and print library and data set member lists
- Move, copy, lock, promote, and print library and data set members
- Reset statistics for ISPF library members
- Define commands to be used with specific dialogs
- Convert SPF menus and messages to ISPF format
- Compare data sets and search for strings of data.

Dialog Test, Foreground, and Batch

PDF provides special facilities for testing dialogs. The Dialog Test option allows testing of individual dialog elements and complete dialogs.

After a program has been developed, you can either assemble it or compile and link it by using either the Foreground or Batch option. The Foreground option gives you the advantage of watching the program being compiled. The advantage of using the Batch option is that it frees the terminal for other uses during the compilation of a lengthy program. If the program contains errors, you can use the debugging capabilities of the Dialog Test facility to help correct them.

TSO Commands and CLISTs

While using PDF, you may need to invoke TSO commands or CLISTs. PDF lets you do this by:

- Prefixing a command with TSO and entering it from any COMMAND field
- Selecting the Command option and entering the TSO command or CLIST on the panel that is displayed
- Entering a TSO command or CLIST in the line command field on a data set list display (option 3.4) or a member list display using option M.

Library Management Facility

To help you maintain different levels or versions of a library member, PDF provides the Library Management utilities. If the Library Management Facility (LMF) is installed, you can lock the version being edited in a private library to prevent someone else from altering the member. When the member is edited to your satisfaction, it can then be promoted to another group within the library for further development or testing.

Other IBM Program Development Products

You may find it necessary or desirable to invoke another IBM program development product while using PDF. The IBM Products option lets you invoke the Cross System Product/Application Development, Cross System Product/Application Execution, Information/System, COBOL Structuring Facility foreground dialog, and COBOL Structuring Facility background dialog licensed programs without leaving PDF. However, these products must be installed and available for your use.

Primer and Online Tutorial

Learning to use PDF is made easy by the:

- *Primer*, which guides you through the process of developing a sample dialog
- Online help and tutorial facilities that are always available while using PDF.

For example, if you need help filling in the data requested by a PDF utility, you can use the tutorial to help you understand the data entry requirements for that utility.

Interacting with PDF

As an interactive dialog, PDF communicates with you through *panels* and *messages*. Ordinarily, the first panel you see when you enter the PDF command is the ISPF/PDF Primary Option Menu, shown in Figure 1 on page 5. Panels display data and selection lists, and provide fields that you enter information in, such as a data set name or an ISPF/PDF command.

The interaction occurs when PDF responds to the information or command you enter. PDF's response can be to display another panel, show you an information or error message, or simply carry out a command. For more information about how panels and messages are displayed, see "Understanding PDF Panels" on page 7.

One of the most helpful aspects of the communication between you and PDF is the online tutorial. Many topics related specifically to ISPF, such as ISPF system commands, are covered sparingly in this introductory chapter because they are described fully in *Dialog Management Guide* and in the online tutorial. If you need information about using the online tutorial, see Chapter 13, "Tutorial (Option T)."

Invoking PDF

Before you can invoke PDF, you must first log on to TSO. When the READY prompt appears on the screen, enter the command **PDF**, unless your installation has established an alias for PDF, such as ISPF or SPF. If so, enter the name of the alias instead.

Option Selection

You can select an option by typing its number or letter in the OPTION field and pressing the ENTER key; for example:

```
OPTION ====>
```

to select the Edit option.

If you select ISPF Parms, Utilities, Foreground, Batch, Dialog Test, LM Utilities, or IBM Products, you get another menu with a secondary list of options. As you become familiar with PDF, you may want to bypass the second menu by typing your first two selections, separating them with a period, on a primary option menu. For example, entering 3.1 on the ISPF/PDF Primary Option Menu has the same effect as entering 3 on the ISPF/PDF Primary Option Menu and 1 on the Utility Selection Menu.

An even faster way to select an option when you invoke PDF is to bypass both the ISPF/PDF Primary Option Menu and the secondary menu. To do this, include your selection as a parameter of the PDF command. For example, here are two selections you can choose and a description of each:

PDF 2 To go directly to the Edit option.
PDF 3.1 To go directly to the Library utility (3.1).

Note: If your installation uses an alias for the PDF command, substitute the alias for PDF in the examples above.

ISPF/PDF Primary Options

The first panel is the ISPF/PDF Primary Option Menu, shown in Figure 1.

```

----- ISPF/PDF PRIMARY OPTION MENU -----
OPTION  ====> _

0  ISPF PARMS  - Specify terminal and user parameters
1  BROWSE      - Display source data or output listings
2  EDIT        - Create or change source data
3  UTILITIES   - Perform utility functions
4  FOREGROUND  - Invoke language processors in foreground
5  BATCH       - Submit job for language processing
6  COMMAND     - Enter TSO command or CLIST
7  DIALOG TEST - Perform dialog testing
8  LM UTILITIES- Perform library administrator utility functions
9  IBM PRODUCTS- Additional IBM program development products
C  CHANGES    - Display summary of changes for this release
T  TUTORIAL    - Display information about ISPF/PDF
X  EXIT        - Terminate ISPF using log and list defaults

Enter END command to terminate ISPF.
```

Figure 1. ISPF/PDF Primary Option Menu

When you select one of these options and press the ENTER key, PDF displays another panel that is determined by the option you selected.

Option Selection

Chapters 3 through 13 describe those options, which are listed in Figure 1. You can use the options to:

- Display and change selected ISPF parameters, such as terminal characteristics and program function (PF) keys. See Chapter 3, “ISPFParms (Option 0)” for more information.
- Display data without changing it and to see large data sets, such as compiler listings. You can scroll Browse displays up, down, left, or right. Browse commands, entered in the COMMAND field, let you do tasks like finding a character string. See Chapter 4, “Browse (Option 1)” for more information.
- Create or change source data, such as program code and documentation. The ISPF editor is a full-screen editor. Unlike Browse, it lets you type over the data being displayed. You can scroll the data up, down, left, or right. You can change the data by using *Edit line commands*, which are entered directly on the line number of the line or lines to be affected, and *primary commands*, which are entered in the COMMAND field. See Chapter 5, “Edit (Option 2)” and *Edit and Edit Macros* for more information.
- Carry out library and data set maintenance tasks, such as moving, copying, or promoting library or data set members, displaying or printing data set names and volume table of contents (VTOC) information, comparing data sets, and searching for strings of data. See Chapter 6, “Utilities (Option 3)” for more information.
- Invoke IBM language processing programs in the foreground. See Chapter 7, “Foreground (Option 4)” for more information.
- Invoke IBM language processing programs as batch jobs. ISPF generates Job Control Language (JCL) for the job based on information you enter on the batch processing panels and submits the job for execution. See Chapter 8, “Batch (Option 5)” for more information.
- Invoke TSO commands or CLISTs under ISPF, to wrap a long command to the next line, and to enter Session Manager mode. See Chapter 9, “Command (Option 6)” for more information.
- Test individual ISPF dialog components, such as panels and messages, and dialog functions, such as programs, commands, and menus. See Chapter 10, “Dialog Test (Option 7)” for more information.
- Control and track the movement of members into controlled libraries and throughout library hierarchies. See Chapter 11, “LM Utilities (Option 8)” and *Library Management* for more information.
- Select other IBM program development products besides ISPF/PDF that are installed and available on your system. The other IBM products that are supported are the Cross System Product/Application Development (CSP/AD), the Cross System Product/Application Execution (CSP/AE), the Information/System (INFO/SYS), the COBOL Structuring Facility foreground dialog (COBOL/SF-F), and the COBOL Structuring Facility background dialog (COBOL/SF-B) licensed

programs. See Chapter 12, “IBM Products (Option 9)” for more information.

- Get online instruction about the changes between Version 1 and Version 2 of ISPF/PDF.
- Get online instruction in the use of ISPF/PDF. You can access the tutorial from other PDF options by using the HELP command. See Chapter 13, “Tutorial (Option T)” for more information.
- Leave PDF using the log and list defaults. You can change these defaults with the Log/List option (0.2).

Understanding PDF Panels

A *panel* is a predefined display image that you see on a display terminal. PDF formats all panels to fit on a 24-line by 80-character screen. On a 3278 Model 3 or 4, data that you can scroll occupies the full length of the screen (32 or 43 lines). On a 3278 Model 5, PDF normally displays information in *default mode*, 24 lines by 80 characters, with the same size characters as other models. Browse and Edit data that is wider than 80 characters is displayed with the smaller *native mode* characters, up to 132 per line. You can use the Terminal option (0.1) to override the automatic switching of modes.

Panel Format

Here is how PDF formats the first three lines of each display:

```
TITLE ----- SHORT MESSAGE
COMMAND/OPTION ----> _ SCROLL ----> PAGE
LONG MESSAGE
```

Figure 2. Panel Format

Understanding PDF Panels

The fields on Figure 2 on page 7 are:

TITLE Identifies the function being carried out and, where appropriate, the library or data set information.

SHORT MESSAGE

Indicates:

- Current line for Browse, and column positions for Browse and Edit
- Current row position in a table display if the SHORT MESSAGE area is not overlaid by a function-requested message
- Successful completion of a processing function
- Error conditions, accompanied by an audible alarm if one is installed. See *Dialog Management Guide* for information about coding an alarm on a panel.

COMMAND/OPTION

Used to enter a command. On a menu, it can be used to enter either a command or an option.

Note: You can place the COMMAND/OPTION field at the bottom of a logical screen and move the LONG MESSAGE line to the line directly above the repositioned COMMAND/OPTION line. See "Display (Option 0.4)" on page 74 for more information.

SCROLL Contains the current scroll amount whenever scrolling applies. You can type over it to change it. The valid scroll amounts are 1 to 9999, PAGE, HALF, MAX, CUR, and DATA.

LONG MESSAGE

Used to display an explanation of error conditions when you enter the HELP command, which is typically done by pressing the PF1/13 key (see "Getting Help" on page 20). On some displays, this line contains column headings or data that is temporarily overlaid by a long message.

Panel Types

When using PDF, you see three basic types of panels:

- Menus (selection panels)
- Data entry panels
- Scrollable data displays.

Menus

A menu, or selection panel, is a panel on which you type a *code* in the **OPTION** field and press the **ENTER** key to select one of the listed items. The code is normally a single digit or letter, which can be typed in either upper- or lowercase. Allowable codes are shown in high intensity. You can also enter ISPF commands. Figure 1 on page 5 is an example of a menu.

*Note: If the code **BLANK** or **blank** is listed, leave the **OPTION** field blank and press the **ENTER** key to select that option. Do not type the word "blank."*

Data Entry Panels

A data entry panel is a panel on which you specify information, such as data set names, job statement parameters, and language processing options. The fields that you can enter information in are labeled and preceded by an arrow (==>). If you fail to enter a required value or if you enter inconsistent values, PDF prompts you with a message. Figure 30 on page 85 is an example of a data entry panel.

Several fields on data entry panels retain their previous values. If the values are correct the next time you use the panel, you do not have to type them again. Just press the **ENTER** key. However, if the values are not correct, type over the values that need to be changed before pressing the **ENTER** key.

The retained values come from your user profile, which ISPF automatically builds and maintains across sessions. See "Remembering Information with User Profiles" on page 28 for more information about user profiles.

Edit modes and defaults are also maintained in the Edit portion of your user profile. See *Edit and Edit Macros* for more information.

Scrollable Data Displays

A scrollable data display is used when all the available data may not fit on the logical screen. The data is displayed in *scrollable format*, which means you can change your view of it by using the scroll commands. You can scroll up and down by using the **UP** and **DOWN** scroll commands. If the data is too wide to fit on the logical screen, you can scroll left and right by using the **LEFT** and **RIGHT** scroll commands.

PDF uses two types of scrollable formats:

- Table format

For member selection lists and certain dialog testing functions, the information is displayed in table format. That is, the information is arranged in rows and columns. Figure 12 on page 40 is an example of a member list.

- Data set format.

Information displayed in data set format is shown exactly as it was typed on the screen. For example, when you browse or edit a data set that contains lines of program code, the lines appear exactly as they were entered. Figure 31 on page 86 is an example of a Browse data display.

Entering Commands in PDF

PDF's flexibility is proved by the various types of commands it accepts and by the many methods that are available for entering these commands. The following table provides a general overview of the types of commands available when using PDF and the methods of entering each type:

Entry Methods	TSO/CLIST Commands	ISPF Commands	PDF Primary Commands	PDF Line Commands
TSO Command Processor panel	X			
OPTION field	X	X		
COMMAND field	X	X	X	
Line command fields (1 to 6 characters)				X
Line command fields (9 characters)	X			X

Figure 3. Command Types and Entry Methods

Two command entry methods not shown in the table are PA and PF keys. PA1 and PA2 are used to issue the ATTENTION and RESHOW system commands, respectively. These are hardware keys that you cannot redefine. The PF keys can be used to enter all four of the command types shown in the table.

The following sections provide basic information about the types of commands and methods of entry.

Types of Commands

There are three types of commands that operate at three different levels:

- TSO commands and CLISTs

At the highest level are commands that you invoke through the operating system, here MVS/TSO. These commands include TSO commands, CLISTs, and the commands assigned to the PA1 and PA2 keys.

- ISPF commands

At the second level are commands that you invoke through PDF's prerequisite licensed program, ISPF.

- PDF primary and line commands.

At the lowest level are commands that you invoke through the PDF licensed program itself. These commands include both primary commands and line commands.

TSO Commands and CLISTs

PDF gives you access to the MVS/TSO operating system by letting you enter TSO commands and CLISTs from within PDF. *TSO/E Command Language Reference* and *MVS/XA TSO/E TSO Command Language Reference* contain descriptions of all TSO system commands.

You can enter most TSO commands and CLISTs always and from all three of the panel types—data entry, menu, and scrollable data display—discussed under “Panel Types” on page 8. However, some TSO commands, such as LOGON and LOGOFF, can cause unwanted results when entered from within PDF. This is also true of CLISTs that contain these TSO commands. See “Rules for Entering TSO Commands” on page 276 and “Rules for Entering CLISTs” on page 277 for information about rules governing the entry of TSO commands and CLISTs from within PDF.

ISPF Commands

As PDF's prerequisite program product, ISPF provides commands that complement the functions and options provided by PDF. Most of these commands are valid always and from all three types of panels.

However, the validity of some commands depends on the type of panel displayed or the type of terminal being used. For example, the scroll commands--UP, DOWN, LEFT, and RIGHT—are valid only on scrollable data displays, and the SPLITV command is valid only if you are using a 3290 display terminal.

ISPF Commands Assigned to PF Keys:

ISPF assigns 12 of its commands to the PF keys. For terminals with 24 PF keys, these 12 commands are repeated for PF13-PF24.

PDF does not change the ISPF command assignments. However, if you want to change them, you can do so by using either the ISPF KEYS command or the PF Keys option (0.3).

The following tables review the default ISPF PF key assignments and the other ISPF commands. See “Program Function (PF) Keys” on page 19 and *Dialog Management Guide* for more information.

PF Keys and Default Commands	Command Definitions
PF1/13 HELP	Displays additional information about an error message or provides tutorial information about ISPF/PDF commands and options. See "Getting Help" on page 20 for more information.
PF2/14 SPLIT	Splits the display screen into two logical screens or changes the location of the split line. See "Splitting and Swapping Display Screens" on page 21 for more information.
PF3/15 END	Ends the current operation and returns to the previous panel. If the ISPF/PDF Primary Option Menu is displayed, this command ends PDF. See "Ending PDF or a PDF Function" on page 22 for more information.
PF4/16 RETURN	Causes an immediate return to the ISPF/PDF Primary Option Menu or to the panel from which a HELP or KEYS command was entered. When entered from the ISPF/PDF Primary Option Menu, it has the same effect as using the Exit option (X). See "Ending PDF or a PDF Function" on page 22 for more information.
PF5/17 RFIND	Repeats the previous FIND command or the "find" part of the most recent CHANGE command. See "Repeating a FIND or CHANGE Command" on page 25 for more information.
PF6/18 RCHANGE	Repeats the previous CHANGE command. See "Repeating a FIND or CHANGE Command" on page 25 for more information.
PF7/19 UP	Causes a scroll toward the top of the data. See "Scrolling Within a Data Display" on page 25 for more information.
PF8/20 DOWN	Causes a scroll toward the bottom of the data. See "Scrolling Within a Data Display" on page 25 for more information.
PF9/21 SWAP	Moves the cursor to wherever it was previously positioned on another logical screen when the screen is split. See "Splitting and Swapping Display Screens" on page 21 for more information.
PF10/22 LEFT	Causes a scroll toward the left margin of the data. See "Scrolling Within a Data Display" on page 25 for more information.
PF11/23 RIGHT	Causes a scroll toward the right margin of the data. See "Scrolling Within a Data Display" on page 25 for more information.
PF12/24 RETRIEVE	Redisplays commands entered in the COMMAND/OPTION field. See "Retrieving Commands" on page 27 for more information.

Figure 4. ISPF Commands Assigned to PF Keys

ISPF Commands	Command Definitions
= <i>option</i> (jump function)	Lets you jump from one option to another without displaying the ISPF/PDF Primary Option Menu or any intervening panels. Unlike the other ISPF commands, the jump function can be used in any input field. Enter an equal sign (=) followed by any option that can be entered on the ISPF/PDF Primary Option Menu.
CURSOR	Repositions the cursor to the COMMAND or OPTION field. If invoked a second time, moves the cursor to the SCROLL field on scrollable data displays. This command is useful only when invoked by a PF key.
ENVIRON	Traces the TPUT, TGET, and PUTLINE buffers, produces system ABEND dumps when not running in ISPF TEST mode, and gathers terminal status information.
ISPPREP	Creates preprocessed panels by converting panel definitions to the internal format used by ISPF, thus improving panel-to-panel transition time.
KEYS	Displays a panel that lets you change your PF key definitions.
LIST	Allows processing of the list data set at any time.
LOG	Allows processing of the log data set at any time. You cannot issue the LOG command while browsing the log by using option 7.5.
PANELID	Displays the name of the current panel in the first eight characters of line one.
PFSHOW	Displays current PF key settings at the bottom of the screen.
PRINT	Copies an image of the <i>display</i> screen into the ISPF list data set.
PRINT-HI	Copies an image of the <i>display</i> screen into the ISPF list data set and causes intensified characters to be printed over to simulate the intensity of the display.
PRINTG	Prints a panel and its graphic area on a GDDM printer if the GDDM Graphic Interface has been initialized.
PRINTL	Copies an image of the <i>logical</i> screen into the ISPF list data set.
PRINTLHI	Copies an image of the <i>logical</i> screen into the ISPF list data set and causes intensified characters to be printed over to simulate the intensity of the display.
SPLITV	Divides the display, on a 3290 terminal in partition mode, vertically into two logical screens that can be used independently.

Figure 5. Other ISPF Commands

Entering Commands in PDF

PDF Primary and Line Commands

PDF primary and line commands are valid only when invoked within PDF. The PDF primary option that you are using determines which commands are valid for that option. For example, the CAPS primary command is valid only when using the Edit option.

You should refer to the chapter in this book that discusses the option you are using for information about the valid PDF primary and line commands for that option. Exceptions to this rule are the primary and line commands for Edit (option 2), which are defined in *Edit and Edit Macros*, and those for LM Utilities (option 8), which are defined in *Library Management*.

PDF primary commands can affect a complete data set or PDS member. On the other hand, PDF line commands can affect only one or more specified lines within a data set. The complete data set is not affected unless you specify all the lines in the data set. For example, the Edit COPY command can copy a complete data set, whereas the C or CC Edit line commands can copy only lines or blocks of lines within a data set.

Methods of Entering Commands

Each type of command has at least two methods of entry. For example, ISPF commands can be typed in either the COMMAND or OPTION field, or they can be assigned to PF keys. PDF provides the following methods for entering all *types* of commands:

- Typing commands on the:
 - TSO Command Processor panel (option 6)
 - COMMAND and OPTION fields
 - Line command fields.
- Pressing program access (PA) and program function (PF) keys.

TSO Command Processor Panel (Option 6)

The TSO Command Processor panel has only one purpose: it allows you to invoke MVS/TSO system commands and CLISTs or ISPF commands without leaving PDF. This panel cannot be used to invoke PDF commands. See Chapter 9, “Command (Option 6)” for information about using this panel.

COMMAND and OPTION Fields

Of the three types of panels described in “Panel Types” on page 8, two—data entry panels and scrollable data displays—provide a COMMAND field. The sole purpose of this field is to let you enter one or more TSO commands or CLISTs, ISPF commands, or PDF primary commands. Commands entered in the COMMAND field are invoked when you press the ENTER key.

The third panel type—menus—provides an OPTION field, which serves a dual purpose. As its name implies, this field is used when you select one of the options displayed on a menu, as explained in “Option Selection” on

page 5. However, you can also use the OPTION field to enter TSO commands and CLISTs, or ISPF commands, but not PDF primary commands.

TSO Commands and CLISTs:

Except for a few cases, almost any TSO command or CLIST can be entered in any COMMAND or OPTION field. See “Rules for Entering TSO Commands” on page 276 and “Rules for Entering CLISTs” on page 277 for more information.

When entering a TSO command or CLIST in a COMMAND or OPTION field, you must enter **TSO** before the TSO command or CLIST name. For example, to use the TSO ALLOCATE command you could enter:

```
COMMAND ===>
```

Also, CLIST names can be preceded by a percent (%) symbol, as in:

```
OPTION ===>
```

This informs TSO that it is a CLIST and not a TSO command.

ISPF Commands:

All ISPF commands can be entered in all COMMAND and OPTION fields and on all types of panels.

PDF Primary Commands:

PDF primary commands cannot be entered in an OPTION field. They can be entered only on a scrollable data display, which can be easily identified by the SCROLL field located at the end of the line that contains the COMMAND field. Examples of scrollable data displays are Edit data displays and member selection lists.

ISPF/PDF Command Notation:

In this book, the following notation conventions are used to describe ISPF and PDF command formats:

- Uppercase commands and their operands should be entered as shown, but not necessarily in uppercase. Operands shown in lowercase are variable; you substitute your own value for them.
- Operands shown in brackets ([]) are optional, with a choice indicated by the OR symbol (|) or by stacking the operands. You can choose one or none; the defaults are underscored.
- Operands shown without brackets are required. If several required operands are separated by the OR symbol (|) or are stacked and shown in braces ({ }), you must select one of the choices.
- Command name truncations are shown stacked under the full command name without braces.

Stacking Commands:

You can enter more than one command on a COMMAND or OPTION field by using a method called *stacking*. To stack two or more commands, enter a special delimiter between them. The default delimiter is a semicolon, but you can change it by using the Terminal option (0.1).

In the following example, two Edit CHANGE commands have been stacked:

```
COMMAND ===>
```

Dual Command Processing:

You can enter information in a COMMAND or OPTION field in combination with pressing a PF key, but only if the information is related to the command assigned to the PF key. The command is invoked first. For example, UP is the default ISPF command assigned to PF7. Here, typing 4 in the COMMAND field and pressing PF7 is the same as typing UP 4 in the COMMAND field and pressing the ENTER key.

If the command you type in the COMMAND field is unrelated to the command assigned to a PF key you have pressed, PDF passes the entry to the function in control. Then, the function in control can either process or ignore the entry. For example, if the Edit option is the function in control, the command in the COMMAND field is processed and PDF may display an error message. However, if the tutorial is in control, the command is ignored. PDF executes any stacked valid commands.

Line Command Fields

In PDF, line command fields can take many forms. Some have headings while others do not. Though most are blank, some contain one or more single quotation marks while others contain sequence numbers. Also, some allow you to enter only one character, while others let you enter up to nine characters and even type over data set names. The following table shows the characteristics of the various line command fields and the functions that use them.

Functions Providing Line Command Fields	Heading Displayed	Characters Allowed	Initial Contents
Member Selection List	None (see 1 on page 17)	1 9	Blank
Edit (Option 2)	None	6	Quotes/ numbers (see 2 on page 17)
Data Set List utility (Option 3.4)	COMMAND	9 (see 3 on page 17)	Blank

Figure 6 (Part 1 of 2). Line Command Field Characteristics

Functions Providing Line Command Fields	Heading Displayed	Characters Allowed	Initial Contents
Command Table utility (Option 3.9)	None	4	Quotes
Format Specification utility (Option 3.11)	None	1	Blank
SCRIPT/VS (Option 4.9)	Line Cmd	1	Quotes
Dialog Test (Option 7): Variables (Option 7.3) Tables (Option 7.4) Traces (Option 7.7) Breakpoints (Option 7.8)	None	4	Quotes

Figure 6 (Part 2 of 2). Line Command Field Characteristics

The following items explain differences referred to in the preceding table:

1. No heading for member list line command fields

The line command field for member selection lists is blank and has no heading. This includes normal member lists, which have a one-character line command field, and the member list displayed when you use option M of the Data Set List utility, which has a nine-character line command field.

2. Line command field content using Edit (option 2)

Quotes appear in the line command field when you create a new data set or member, or when you insert one or more lines. Sequence numbers appear if NUMBER ON is set in your Edit profile.

3. Ability to type over data set names

Unlike the other functions that use line command fields, data set lists with nine-character line command fields allow you to type over data set names. This capability extends the length of these line command fields, allowing you to enter long TSO commands and CLIST names.

TSO Commands and CLISTs:

Most line command fields allow only the entry of line commands. However, data set lists and member lists that are displayed using the Data Set List utility (option 3.4) provide nine-character line command fields. These fields can be used to enter TSO commands and CLISTs. See "TSO Commands and CLISTs" on page 132 for information about entering TSO commands and CLISTs in a line command field. Also, "Rules for Entering TSO Commands" on page 276 and "Rules for Entering CLISTs" on page 277 contain information about the rules governing the entry of TSO commands and CLISTs from within PDF.

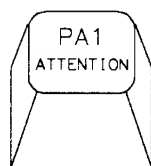
PDF Line Commands:

Most PDF line commands use only one letter, such as “S” for selecting a member from a member list. Others, such as many of the Edit line commands, use more than one letter and, sometimes, let you add a number so the command will affect more than one line. For example, the UC line command plus the number 3, as in UC3, converts three lines to uppercase.

Another type of line command is the *block* line command, which affects the block of lines between and including the lines on which the commands are entered. For example, the UCC line command, when typed beside two different lines, converts all lines between and including the two commands to uppercase.

Program Access (PA) Keys

The definitions assigned by MVS to the two PA keys cannot be changed. The PA keys are:



ATTENTION

The ATTENTION key (PA1) interrupts PDF processing. This key is most commonly used when the display screen has been split.

Note: See “Splitting and Swapping Display Screens” on page 21 if you need information about splitting the display screen.

Normally, you should not use this key while you are in ISPF full screen mode.

If you press PA1 after the keyboard is unlocked by ISPF, you are shown the last full screen image displayed by ISPF again, just as if you had pressed PA2. It does not cause you to leave ISPF. However, if you press the PA1 key a second time without any intervening interaction, current processing stops and the ISPF/PDF Primary Option Menu is displayed again.

If you press the PA1 key after manually unlocking the keyboard (with the RESET key), current processing is stopped and the ISPF/PDF Primary Option Menu is displayed again.



RESHOW

The RESHOW key (PA2) shows the last full screen image displayed by ISPF again. PA2 can be useful if you accidentally press the ERASE INPUT or CLEAR key, or if you type unwanted information but have not yet pressed the ENTER key or a PF key.

Note: PDF does not allow use of the Field Mark character, which is the same key as PA2.

Program Function (PF) Keys

You can enter often-used commands faster by assigning them to the program function (PF) keys instead of typing them and pressing the ENTER key. No PF keys are required; they simply make it easier to enter commands.

Examples of keys shown in this section are based on the keyboard for the 327x terminals. If you are not using a 327x terminal, these examples may not match your keyboard layout, but the function of the corresponding keys is the same.

If you are using a 3275 or 3277 terminal, do not press the RESET key during PDF processing and try to enter additional information or use a PF key. This action can produce unwanted results. Instead, wait until processing has completed.

Default ISPF Command Assignments:

Twelve ISPF commands are initially assigned to the PF keys as defaults. The default PF key command settings are shown in Figure 7 on page 20. See "Using the Default PF Key Settings" on page 20 for information about using these PF key settings.

If the terminal you are using has 24 PF keys, the same commands that are assigned to PF keys 13-24 are also assigned to PF keys 1-12. The HELP command is assigned to PF1/13, the SPLIT command to PF2/14, and so forth. However, there are 12 additional ISPF commands, plus TSO commands, CLISTs, and PDF primary and line commands that you may want to assign to these keys instead. You can assign any of these command types to PF keys by using the PF Keys option (0.3) or the KEYS command. See "ISPF Commands" on page 11 for tables that provide the names and brief definitions of all 24 ISPF commands.

Using the Default PF Key Settings

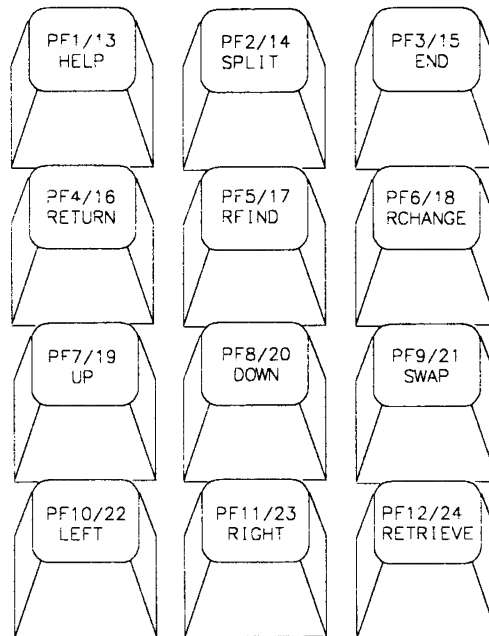


Figure 7. Default PF Key Settings

Using the Default PF Key Settings

The following sections supplement the command definitions provided in “ISPF Commands Assigned to PF Keys” on page 11.

Getting Help

The **HELP** command (PF1/13) shows you general information about an ISPF system command, PDF command, option, or panel, or additional information about a message that has been displayed in the upper-right corner of the screen.

If a short message is displayed, entering the **HELP** command displays a one-line explanation of the message. If this explanation is not sufficient, you can get further information by entering the **HELP** command a second time. This causes an entry into the appropriate section of the tutorial.

For some messages in the LM Utilities (option 8), entering the **HELP** command a second time displays a message help panel instead of entering the tutorial. This help panel provides a description of:

- The error condition that would cause the message
- The action needed to correct the error.

If a message is not displayed, the **HELP** command causes a direct entry into the appropriate section of the tutorial. Once in the tutorial, enter the **END** (PF3/15) or **RETURN** (PF4/16) command to return to the screen that you were viewing when you entered the **HELP** command.

Splitting and Swapping Display Screens

This section discusses:

- Dividing the display screen with the SPLIT command (PF2/14)
- Moving the cursor from one logical screen to another with the SWAP command (PF9/21).

Using the SPLIT Command

The SPLIT command (PF2/14) divides the display screen into two logical screens separated by a dotted horizontal line. The line containing the cursor becomes the split line. All lines below the dotted line are treated as the bottom screen.

The first display to appear on the bottom screen is always the ISPF/PDF Primary Option Menu. Figure 8 shows what a screen looks like after it is split:

```
BROWSE -- ISPFDEMO.MYLIB.PLI(COINS) - 01.04 ----- LINE 000000 COL 001 080
COMMAND ==> SCROLL ==> PAGE
***** TOP OF DATA *****
COINS: 00010001
PROCEDURE OPTIONS (MAIN); 00020000
----- ISPF/PDF PRIMARY OPTION MENU -----
OPTION ==> _
0 ISPF PARMS - Specify terminal and user parameters USERID - BECKT
1 BROWSE - Display source data or output listings TIME - 12:47
2 EDIT - Create or change source data TERMINAL - 3278
3 UTILITIES - Perform utility functions PF KEYS - 12
4 FOREGROUND - Invoke language processors in foreground
5 BATCH - Submit job for language processing
6 COMMAND - Enter TSO command or CLIST
7 DIALOG TEST - Perform dialog testing
8 LM UTILITIES- Perform library administrator utility functions
9 IBM PRODUCTS- Additional IBM program development products
C CHANGES - Display summary of changes for this release
T TUTORIAL - Display information about ISPF/PDF
X EXIT - Terminate ISPF using log and list defaults

Enter END command to terminate ISPF.
```

Figure 8. Split Screen Example

Repositioning a Split Screen:

You can reposition the split screen by moving the cursor to the new location and using PF2/14 or the SPLIT command again. For example, you may need to reposition the split location to bring a complete panel or tutorial page into view.

Split Screen Limitations:

PDF treats the top and bottom screens as though they were on separate terminals. Functions that can be done simultaneously on two separate terminals can be done on the top and bottom screens. However, functions that are in conflict on two separate terminals are also in conflict when you try them simultaneously on the split screen, such as editing the same sequential data set or executing the same application program.

Using the Default PF Key Settings

Also, only one logical screen is active at a time. Therefore, while executing a foreground compilation or a TSO command, you cannot do any other function until the compilation or command has finished processing. This includes splitting the display screen and, if the screen is already split, “swapping” the cursor from one logical screen to the other.

Using the SWAP Command

To switch the cursor from one screen to the other, use the cursor movement keys or the SWAP command (PF9/21). The location of the cursor identifies which of the two screens is active. Any interrupts, such as pressing the PA1 key or a PF key, affect only the active screen.

Also, the SWAP command automatically repositions the split if either logical screen is less than five lines long. This allows each of the two logical screens to consume nearly all the physical screen. For example, if the split occurs at the third line down from the top and the cursor is on the bottom screen, SWAP moves the cursor to the top screen and repositions the split to the third line up from the bottom.

Notes:

1. *You cannot reposition the split by moving the cursor, unless you enter the SPLIT command after moving the cursor.*
2. *Each time you enter the SWAP command, the cursor returns to the position it occupied before the previous “swap.”*

To return to a full screen display, move the cursor to the logical screen you want to remove. Then end PDF on that screen by using the END command (PF3/15), the RETURN command (PF4/16) or the Exit option (X).

Ending PDF or a PDF Function

To end a PDF function, such as Edit, without ending PDF, just press PF3/15 or PF4/16. The result is the same as typing **END** or **RETURN**, respectively, on the COMMAND line and pressing the ENTER key.

Notes:

1. *If you are using the Hardcopy utility (option 3.6), the END and RETURN commands submit your job for processing. Enter the CANCEL command to leave the Hardcopy utility without submitting a job.*
2. *If a SuperC or Search-For member list is displayed, the END command processes any members you have selected. Enter CANCEL or RETURN to leave one of these member lists without processing your selections.*

To end PDF from any panel, except when you are using Dialog Test (option 7), use the jump function with option X by entering =X. If you are using a Dialog Test option, entering =X returns you to the ISPF/PDF Primary Option Menu; if you are at the Dialog Test Primary Option Menu, entering either X or =X returns you to the ISPF/PDF Primary Option Menu.

To end PDF from the ISPF/PDF Primary Option Menu, you can use the:

- END command (PF 3/15)
- RETURN command (PF 4/16)
- Exit option (X).

If the display screen is split, taking one of the actions listed above ends PDF on the active logical screen only. See “Splitting and Swapping Display Screens” on page 21 for more information.

Using the END Command

The END command (PF3/15), when entered from the ISPF/PDF Primary Option Menu, displays the Specify Disposition of Log and List Data Sets panel (Figure 9) if an ISPF log or list data set has been created. This panel lets you specify how to process the log and list data sets. If you have already specified log and list defaults on the Log and List Defaults panel (option 0.2), the Specify Disposition of Log and List Data Sets panel contains these defaults.

```
----- SPECIFY DISPOSITION OF LOG AND LIST DATA SETS -----
COMMAND ---->

LOG DATA SET DISPOSITION                                LIST DATA SET DISPOSITION
-----
Process option  ----> _                                     Process option  ---->
SYSOUT class   ----> A                                     SYSOUT class   ----> A
Local printer ID ---->                                     Local printer ID ---->

VALID PROCESS OPTIONS:
  PD - Print data set and delete
  D  - Delete data set without printing
  K  - Keep data set (allocate same data set in next session)
  KN - Keep data set and allocate new data set in next session

Press ENTER key to complete ISPF termination.
Enter END command to return to the primary option menu.

JOB STATEMENT INFORMATION: (Required for system printer)
----> //FRANKEL JOB (ACCOUNT),'NAME'
----> //*
----> //*
----> /*
```

Figure 9. Specify Disposition of Log and List Data Sets Panel

The panel shown in Figure 9 is the version that is displayed if both log and list data sets have been used. If only one has been used, a version of the panel is displayed for only that one. If neither list nor log has been used, PDF ends immediately when you enter either the Exit option (X), the END command, or the RETURN command from the ISPF/PDF Primary Option Menu.

Note: The LOG and LIST commands let you process log and list data sets at any time, not just when you leave PDF.

Using the Default PF Key Settings

Only one field on the Specify Disposition of Log and List Data Sets panel always requires an entry:

Process option

Tells PDF how to process the list and/or log data sets. The valid process options are:

PD	Print the data set and then delete it.
D	Delete the data set without printing it.
K	Keep the data set and allocate it in the next session.
KN	Keep the data set, but allocate a new data set in the next session.

If you want to print the log or list data set, or both, you can specify the following information:

SYSOUT class

Destination of printed data set. Required only if the data set is to be printed and local printer ID is not specified.

Local printer ID

Destination of printed data set. Required only if data set is to be printed and SYSOUT class is not specified.

JOB STATEMENT INFORMATION

TSO Job Control Language (JCL) that is required for batch printing and language processing. See "Job Statement Information" on page 55 for more information.

When you type the information, press the ENTER key to process the log/list data sets and end PDF. The screen clears, then one or more messages appear and you return to TSO.

Using the RETURN Command

The RETURN command (PF 4/16) causes an immediate return to the ISPF/PDF Primary Option Menu or to the panel from which a HELP or KEYS command was entered, without displaying any intervening panels.

Note: When you are using the Dialog Test option (7), the RETURN command stops at the Dialog Test Primary Option Menu.

When entered from the ISPF/PDF Primary Option Menu, the RETURN command has the same effect as using the Exit option (X). The Specify Disposition of Log and List Data Sets panel is not displayed unless both of the following are true:

- No defaults are specified on the Log and List Defaults panel (option 0.2).
- A log and/or list data set is generated.

Using the Exit Option (X)

The Exit option ends PDF using any defaults for processing log and list data sets that you have specified with the Log/List option (0.2). If you have not specified any defaults, the Exit option displays the Specify Disposition of Log and List Data Sets panel (Figure 9 on page 23), but only if a log and/or list data set has been generated.

If you are not in split-screen mode, you can use the *jump function* to immediately leave PDF from any panel by entering =X, with the following exception:

- If you are using any of the Dialog Test options (7.1 through 7.T), entering =X returns you to the ISPF/PDF Primary Option Menu; if you are at the Dialog Test Primary Option Menu, enter either X or =X to return to the ISPF/PDF Primary Option Menu. Here, either “x” or “=x” ends PDF.

Repeating a FIND or CHANGE Command

The RFIND and RCHANGE commands, assigned to PF5/17 and PF6/18, respectively, are used to repeat the:

- FIND command (Browse, Edit, and Data Set List)
- CHANGE command (Edit only).

Using the RFIND Command

After entering the FIND command, you can press PF 5/17 to repeat this command without having to type it on the COMMAND line again.

Using the RCHANGE Command

The CHANGE command has to carry out a “find” operation before it can make a change. Therefore, after entering the CHANGE command, you can:

- Press PF6/18 to repeat the CHANGE command.
- Press PF5/17 to repeat the “find” part of the CHANGE command.

After pressing PF5/17, if you want to make the change, you can press PF6/18. If you do not want to make the change, you can press PF5/17 again to find the next occurrence.

Scrolling Within a Data Display

During Edit, Browse, table display, member list display, and data set list display, you are usually dealing with information that exceeds the screen size. Scrolling allows you to move the screen *window* up and down across the information. If the information exceeds the screen width, you can move the screen window left and right.

Using the Default PF Key Settings

Using the Scroll Commands

The scroll commands and PF key settings are:

PF7/19 - UP	Scrolls toward the top of the data.
PF8/20 - DOWN	Scrolls toward the bottom of the data.
PF10/22 - LEFT	Scrolls toward the left margin of the data.
PF11/23 - RIGHT	Scrolls toward the right margin of the data.

In Edit, the current setting of the bounds affects the LEFT and RIGHT scroll commands. See the description of the BOUNDS primary command in *Edit and Edit Macros* for more information.

Typing Over Scroll and Multiple-Choice Parameters

If you assign the CURSOR command to a PF key, pressing that PF key when the cursor is in the data display area moves the cursor to the COMMAND/OPTION field, regardless of where it is located. Pressing it a second time moves the cursor to the second input field on that line, if any. This is the SCROLL field on panels where scrolling is allowed. If a second input field does not exist, the cursor does not move from the COMMAND/OPTION field.

The SCROLL field value tells PDF how many lines or columns to move each time you use a scroll command. To change the scroll amount, move the cursor to the SCROLL field and type over the amount shown.

If you are browsing or editing a formatted data set, scrolling left or right is done on a field basis, not a column basis. If you specify a scroll amount, PDF ignores it. Instead, scrolling left places the preceding field in the record at the leftmost edit area position; scrolling right places the next field in the record at the leftmost edit area position. See "Using Format Definitions" on page 50 for more information about browsing or editing formatted data sets.

Valid scroll amounts are:

1 to 9999	To scroll by the specified number of lines or columns.
PAGE	To scroll by one page. In this context, "page" means the number of lines or columns in the scrollable area of the logical display screen, which is usually 21. If the screen is split, the number of lines or columns scrolled depends on where you split the screen.
HALF	To scroll by half a page. See the definition of "page" above.
MAX	To scroll to the top, bottom, left, or right margin.
CUR	To scroll to the position of the cursor within the data. If the cursor is outside the data or is positioned in the top, bottom, left, or right margin, PDF scrolls one page. See the definition of "page" above.

DATA To scroll by a page minus one line or column. See the definition of “page” above.

PDF saves the current scroll amount in your user profile. Four different values are saved: one for Browse, one for Edit, one for member lists, and one for the Data Set List utility (3.4). See “Remembering Information with User Profiles” on page 28 for more information about user profiles.

When you type over the scroll amount, the new value remains in effect for that type of operation until you change it again. The value MAX is an exception; following a MAX scroll, the scroll amount reverts to its previous value.

To reduce keystrokes, you can change the SCROLL amount by typing over the first character(s):

- To change the scroll amount to PAGE, HALF, MAX, CUR, or DATA, type **P**, **H**, **M**, **C**, or **D**, respectively, over the first character(s).
- To change the scroll amount to a specific number of lines or columns, type the number over the first character. PDF ignores any alphabetic characters that follow a number. For example, PDF interprets **3AGE** as **3**.

The same rule applies to multiple-choice parameters when the first letter uniquely defines the option. For example:

```
REPLACE MEMBERS  ===>      (YES or NO)
DISPOSITION      ===>      (KEEP or DELETE)
```

To change these options to YES or DELETE, type over the first character with **Y** or **D**, respectively.

Retrieving Commands

The RETRIEVE command is the default command assigned to PF12/24. Entering the RETRIEVE command causes ISPF to redisplay the last command entered in the COMMAND/OPTION field. It applies only to commands entered in the COMMAND/OPTION field. Commands entered by pressing a PF key are not redisplayed. For example, suppose you type the PAGE command in the COMMAND field and then press PF7/19 (the UP command). If you then enter the RETRIEVE command, only the PAGE command is redisplayed.

When ISPF redisplay a command, the cursor is positioned at the end of the command. You can then:

- Enter the command again.
- Change the command and then enter it again.
- Erase the command.

If you have not entered any commands before entering the RETRIEVE command, the COMMAND/OPTION field remains blank with the cursor in the first position.

Remembering Information with User Profiles

How ISPF Stores Commands

Each time you enter a command or command chain, ISPF stores it in a stack. These commands or command chains can contain up to 255 characters. Your system programmer allocates the stack when ISPF is installed. Therefore, the number of commands or command chains that ISPF can store, and subsequently retrieve, depends on the size of the stack allocated by your system programmer.

Note: ISPF does not store the RETRIEVE command in the stack, even if you enter it in the COMMAND/OPTION field. Also, if you enter the same command several times in succession, ISPF only stores it once.

How ISPF Retrieves Commands

ISPF retrieves commands from the stack in reverse order. The last command you entered is the first command retrieved and redisplayed. If you continue to enter the RETRIEVE command until all commands have been redisplayed, ISPF starts over at the top of the stack the next time the RETRIEVE command is entered.

Remembering Information with User Profiles

PDF “remembers” information on your behalf in your user profile. This allows PDF to insert a value in panel input fields by using the values you last entered on either the same panel or a similar type of panel. Sometimes default values are provided if you have not specified otherwise. Information maintained in your user profile includes:

- Project name, group name(s), and type
- Job statement information
- SYSOUT class for printed output
- Defaults for list and log allocation and processing
- Terminal characteristics and PF key definitions
- Edit profiles, including mask, tabs, and bounds
- Current scroll amount for Browse, Edit, Data Set List, and member lists
- Processing options for each of the language processors
- Data set allocation/information parameters.

This information is automatically maintained from one session to another. If you are a new user, you have to enter certain information the first time it appears. But from that point on, you simply review the information and make whatever changes you want before proceeding.

Nine sets of job statements are retained, one set each for the Hardcopy utility (3.6), the Outlist utility (3.8), the SuperC utility (3.12), the SuperCE utility (3.13), the Foreground Print Options panel (option 4, excluding SCRIPT/VS), the Batch option (5), SCRIPT/VS draft output, SCRIPT/VS final output, and ending PDF. This lets you provide different job statement parameters for each of these functions. For more information on job statements, see “Job Statement Information” on page 55.

CHAPTER 2. LIBRARIES AND DATA SETS

This chapter describes:

- **ISPF libraries:** what they are, how they are organized, and how you use PDF to allocate them. The chapter also describes ISPF library statistics, which give you useful information about each member in an ISPF library.
- **How you specify the ISPF library or TSO data set you want to work with.** This includes information on using passwords to control access to data sets.
- **Member list processing.** A member list allows you to list the names of all members or a subset of members in an ISPF library or other partitioned data set. When a member list is displayed, you can locate a particular member or, depending on the type of member list, select one or more members for processing.
- **List and log data sets.** The list data set helps you obtain hardcopy listings of source modules, and the log data set maintains a record of significant user activities.

To allow you to see how PDF works, a sample PDF session completes the chapter.

Chapter 2. Libraries and Data Sets

An *ISPF library* is a cataloged partitioned data set with a three-level name consisting of a project, a group, and a type. An optional library *member* name can also be included. A member can consist of programming code, data, or text.

PDF displays library names on line 1 of a scrollable data display, such as the member list shown in Figure 12 on page 40. Each library generally contains members with the same type of information.

Before you can create and use a new ISPF library or data set, you must *allocate* it. Libraries and data sets are allocated with option A of the Data Set utility (option 3.2). The first step in the allocation process is naming the ISPF library or data set. The following section explains how to do this. See “Data Set Utility (Option 3.2)” on page 106 for other information about data set allocation.

Naming ISPF Libraries and Data Sets

On data entry panels that require a library or other data set name, such as the Data Set Utility panel (option 3.2) and the Edit - Entry Panel (option 2), two groups of fields are provided: one for entering an ISPF library name and one for entering an “other” partitioned or sequential data set name. The next two sections explain these two groups of fields.

ISPF Library Names

To name an ISPF library, you must specify at least a project, group, and type:

```
ISPF LIBRARY:
  PROJECT ==>
  GROUP   ==>
  TYPE    ==>
```

where:

PROJECT The common identifier for all ISPF libraries belonging to the same programming project. This name must be your user ID unless you are using a specific project name that has been predefined in the MVS master catalog.

Naming ISPF Libraries and Data Sets

GROUP The identifier for the particular set of ISPF libraries; that is, the level of the libraries within the library hierarchy. For example, the group name of your private library could be PRIVATE or perhaps your first name, such as in the example in Figure 10 on page 33.

TYPE The identifier for the type of information in the ISPF library, such as PL/I, SCRIPT, and PANELS.

Each component of the library name can be up to eight alphanumeric characters, the first of which must be alphabetic. This conforms to standard TSO data set naming conventions. For convenience, any cataloged data set (sequential or partitioned) with a three-level name can be entered in the PROJECT, GROUP, and TYPE fields, with one level of the name in each field. This can be done even if the data set is not an ISPF library.

If both a library and a data set name are specified on the same panel, the data set name takes priority. Therefore, when specifying a library, leave the DATA SET NAME field blank.

When the library identification is displayed in a title line or message, the project name, group name, and type name are separated with periods. A member name, if applicable, is enclosed in parentheses. For example:

PROJECT.GROUP.TYPE(MEMBER)

On most data entry panels that allow ISPF library name specification, a MEMBER field is available:

MEMBER ===>

where:

MEMBER

The name of an ISPF library or other partitioned data set member. Leaving this field blank or entering a pattern causes PDF to display a member list. See "Displaying Member Lists" on page 38 for more information.

The ISPF library project, group, and type must always accompany the member name, if one is entered. If you try to edit a member that does not exist, PDF provides an Edit display screen with a blank data area.

Member names entered in the MEMBER field or those enclosed in parentheses and entered in the DATA SET NAME field must follow standard PDF naming conventions. That is, the member name must not exceed eight characters, must begin with an alphabetic or national character, and must contain only alphanumeric or national characters.

Naming ISPF Libraries and Data Sets

If you have a partitioned data set with members whose names do not follow PDF naming conventions, PDF allows limited processing, as follows:

- Browse (option 1) allows any character string as a member name in either the MEMBER or DATA SET NAME field and attempts to browse the specified member.
- In Edit (option 2), an existing member with a non-standard member name can be edited, but you cannot create a member with a non-standard member name.
- In general, PDF can handle most member names, with the following exceptions:
 - Selecting a member name that begins with a blank or has embedded blanks can cause unpredictable results.
 - CLIST processing in both Foreground (option 4) and Batch (option 5) can result in a run-time error if a member name includes one of several special characters, such as an ampersand (&).

Other Partitioned or Sequential Data Set Names

You can use the following field to specify any partitioned or sequential data set:

```
OTHER PARTITIONED OR SEQUENTIAL DATA SET:  
DATA SET NAME ===>
```

where:

DATA SET NAME

Any fully-qualified data set name, such as:

```
DATA SET NAME ===>
```

You can include a TSO user prefix or user ID as the first-level qualifier of the data set name. If you omit them and you have created a TSO user prefix, it is automatically added to the beginning of the data set name. If you omit them and you do not have a TSO user prefix, your user ID is added instead. Also, if you include your user prefix or user ID, you must enclose the data set name with apostrophes. PDF does not support partitioned data sets with record format FBS or VBS.

For partitioned data sets, a member name enclosed in parentheses can follow the data set name, such as:

```
DATA SET NAME ===>
```

Omitting the member name and parentheses or using a pattern causes PDF to display a member list. See “Displaying Member Lists” on page 38 for more information about displaying member lists.

Using LMF to Control ISPF Libraries

You can refer to generation data sets by using a signed or unsigned number in place of a member name. However, you can do this only in the DATA SET NAME field. Here's an example:

```
DATA SET NAME    ==>
```

This example refers to the most recently allocated data set in the generation data group. Minus numbers refer to previously allocated data sets; positive numbers refer to unallocated data sets of the group.

Volume Serials

Along with a data set name, you can optionally specify a volume serial. The system catalog is not used when the volume serial is specified:

```
VOLUME SERIAL    ==>                (If not cataloged)
```

where:

VOLUME SERIAL

A real DASD volume or a virtual volume residing on an IBM 3850 Mass Storage System. To access 3850 virtual volumes, you must also have MOUNT authority, which is acquired through the TSO ACCOUNT command. PDF does not allow the use of data sets that reside on more than one volume.

Using LMF to Control ISPF Libraries

If the Library Management Facility (LMF) is installed, you can use *controlled* libraries to:

- Effectively maintain control over different versions of the same application program
- Reduce the risks of writing over members accidentally and of two or more users changing the same member simultaneously. For more information about LMF, see *Library Management*.

Controlled libraries are established and maintained by a library administrator, who must follow these steps in the order shown:

1. Specify the library controls.

The library administrator creates a *library hierarchy* by using the Controls utility (8.1) to specify controls for a series of related libraries. The administrator can then use this hierarchy to limit access to these libraries.

Each level of the hierarchy can support a separate stage of a development project. Figure 10 on page 33 shows a sample three-level hierarchy consisting of a set of master libraries, a set of test libraries, and three sets of private development libraries identified by user ID.

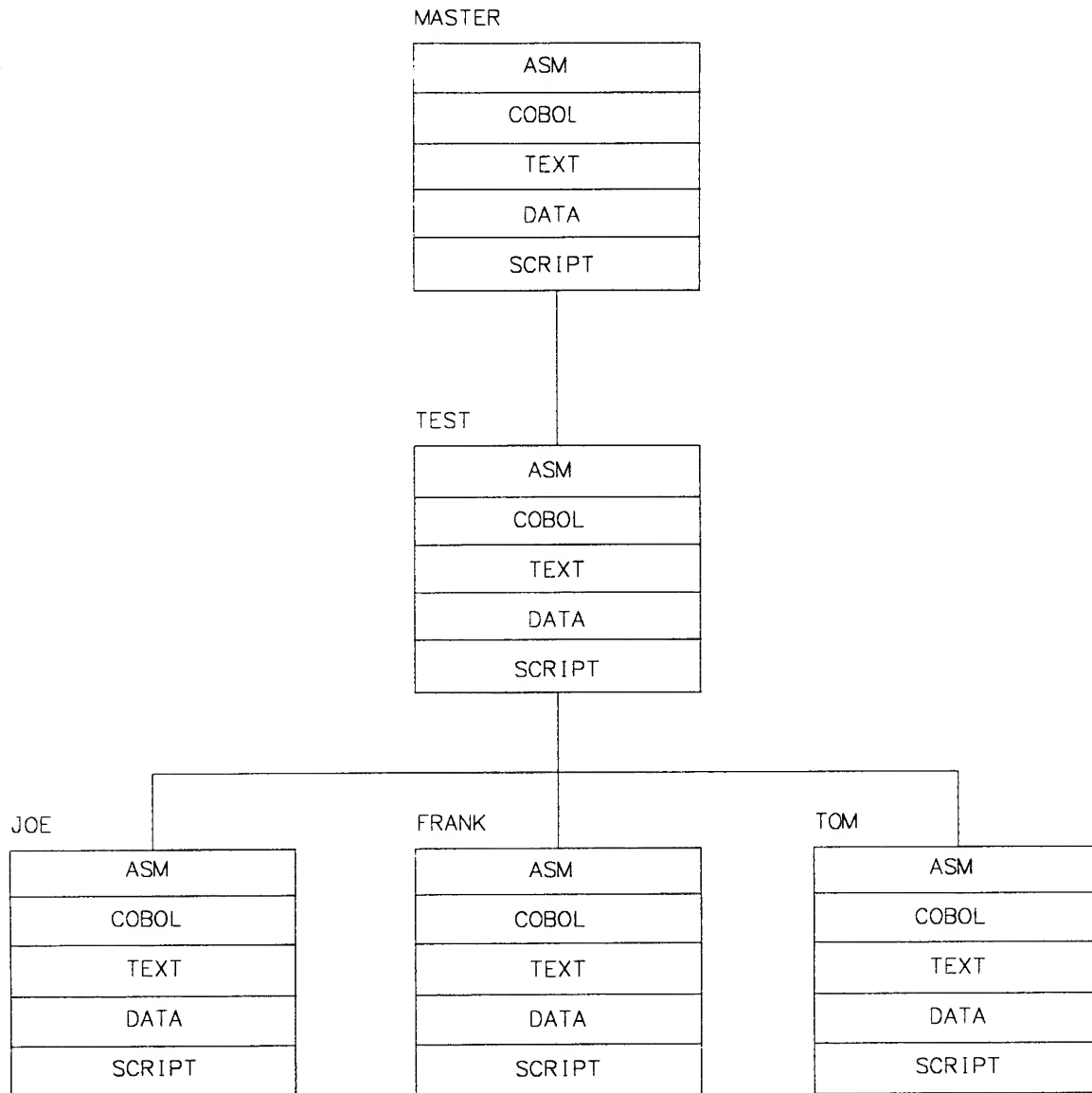


Figure 10. Hierarchy of ISPF Libraries

Typically, a higher degree of control is maintained over the higher-level libraries than over the private development libraries. Libraries can be accessed from the Browse, Edit, Utilities, Foreground, and Batch primary options.

2. Allocate the libraries.

The next step is to allocate the libraries for which controls have been specified, using option A of the Data Set utility (3.2). The Data Set utility is used because the term *library* is merely another name for a *cataloged partitioned data set*.

Each library must be allocated separately because the characteristics of each library, such as record format and logical record length, may vary.

Using LMF to Control ISPF Libraries

PDF automatically catalogs libraries as the library administrator allocates them, giving them a volume serial.

3. Activate the library controls.

The final step is to activate the library controls using the Activate utility (8.4). The controls specified for each library must be activated separately. Once the library administrator has activated these controls, users who have been identified in the LMF control file are free to carry out functions such as browsing, creating, and editing data sets in the library hierarchy, but only within the controls the library administrator has established.

Library Concatenation

The GROUP field is the second of three fields used to enter an ISPF library or other partitioned data set name, the others being PROJECT and TYPE. Whenever the first GROUP field is accompanied by three additional fields horizontally across the screen, you can enter a *library concatenation sequence*, which is a series of group names chained together. PDF searches these groups in the sequence that you enter them in.

Concatenation applies to libraries of the same type and is restricted to libraries that belong to the same project. Generally, the lowest-level library is concatenated ahead of the next higher-level library, and so on, in bottom-to-top order. Therefore, concatenation is usually most effective if this search sequence is the same as the library hierarchy.

For example, new library members or members undergoing changes generally reside in libraries used by program developers. A test library may contain members that have been unit tested and are ready for integration test. A master library might contain fully-tested members, which may correspond to a previously-released version of the program.

Figure 10 on page 33 shows a sample three-level hierarchy consisting of a set of master libraries, a set of test libraries, and three sets of private development libraries identified by user ID. Using this hierarchy, a typical concatenation sequence for project ISPFPROJ, type DATA, and member PGM1 would be:

```
ISPF LIBRARY:
PROJECT ===> ISPFPROJ
GROUP    ===> DATA      ===> TEST      ===> MASTER  ===>
TYPE     ===>
MEMBER   ===> PGM1
```

In this example, the search for member PGM1 would go through libraries:

```
ISPFPROJ.JOE.DATA
ISPFPROJ.TEST.DATA
ISPFPROJ.MASTER.DATA
```

Make sure the concatenated libraries have consistent record formats, logical record lengths, and block sizes. You can use concatenation with the following PDF functions:

- Browsing
- Editing
- Selecting Library utility (3.1) functions:
 - Print index or complete data set
 - Print, rename, delete or browse members
 - Compress data set.
- Compilation
- Assembly
- Link editing
- SCRIPT/VS processing.

Note: You can also use additional input libraries for compilations and assemblies.

Concatenation During Editing

The purpose of concatenation during editing is to provide for copying members to your development library. The concatenation sequence is used to search the libraries for the member to be edited. The edited member is saved in your development library, the first library in the concatenation sequence, while the unchanged version remains in the test or master library. When the new version is fully tested, you can use the Move/Copy utility (3.3) to:

- *Promote* the new version to a higher-level library if the library is controlled by LMF. If you rename the member when you promote it, you must have authority to promote the member under the new name. A member with the old name in the “to” library is neither replaced nor freed. See “Library Member Change and Promotion” on page 36 for more information about promoting library members.
- *Move* the new version to a higher-level library if the library is *not* controlled by LMF.

Concatenation During Language Processing

The purpose of concatenation during language processing is to:

- Help you include source segments in their proper order when using INCLUDE or COPY statements or SCRIPT “imbed” controls
- Allow debugging of new or changed programs without altering the contents of the test or master libraries.

The output from a compilation or assembly (an object module) or from a link edit (a load module) is stored in the lowest-level OBJ or LOAD library, the first library in the concatenation sequence.

Library Member Change and Promotion

To change a member in a controlled library, you must first lock the member in your private library under your logon ID. While a member is locked, no one else can change it in the controlled library. You can lock a member by using the LOCK field on the Edit - Entry Panel (see "Locking Members While Editing") or the L and LP options of the Move/Copy utility (3.3).

When you return a member from a lower-level library to a higher-level library in a hierarchy of controlled libraries, you are *promoting* the member. If a member is locked, you can promote it using the P and PP options of the Move/Copy utility. You can edit a member of a controlled library without locking it. However, you will not be able to promote the edited member to the controlled library afterwards if you do not lock it *before* editing it.

When promoting, you can specify MOVE or FREE. MOVE promotes the member into the specified controlled library and deletes it from your private library. FREE deletes the member from your private library, but does not promote it. Both MOVE and FREE make a locked member available to others.

Locking Members While Editing

A LOCK field is included on the Edit - Entry Panel:

LOCK ==> (YES, NO or NEVER)

where:

LOCK If LMF is installed on your system, you can edit-lock a member of a controlled library that is part of a concatenation sequence, but only if the member does not exist in your private library. Valid values for the LOCK field are:

YES	Tells PDF to edit-lock the member. The member is locked under your logon ID.
NO	Tells PDF <i>not</i> to edit-lock the member, but to change this value to YES for the next Edit session.
NEVER	Tells PDF <i>not</i> to edit-lock the member and to retain this value for future Edit sessions.

Edit-locking is important for two reasons:

- Keeping other users from having access to that member while you are editing it
- Promoting the member back to the controlled library when you have finished editing it. If you do not edit-lock the member, you cannot promote it.

If you save any changes you made while editing the member, it remains locked in your private library. The version of the member stored in the

controlled library remains unchanged until you promote the one in your private library. However, if you leave Edit without saving the changes, they are lost.

Conditions When Locking is Ignored:

PDF ignores the YES value in the LOCK field if:

- The LMF control file (ISRCFIL) is not allocated, which means the LOCK field has no meaning to PDF.
- A member in your private library has the same name as the controlled library member that you want to edit-lock. Here, you can either rename, move, or delete the private library member.

Conditions When Locking Causes Errors:

The following conditions can cause an error if you enter YES in the LOCK field:

- The libraries are not concatenated in the proper sequence.
- The library controls are not active.
- The member you want to edit-lock is locked with another user's logon ID.

Even if an error condition keeps you from locking a member, you can still edit it. Just remember that you cannot promote it back to the controlled library afterwards. PDF displays a panel that gives you a choice between pressing the ENTER key to edit the member or entering the END command if you decide not to edit.

Using Member Selection Lists

A *member selection list*, also called a member list, is initially an alphabetic list of the members of an ISPF library or TSO partitioned data set. The following table provides a quick reference to the primary options that display member lists and their differences. In the **Type of Selection** column, "Single" means PDF processes only the line command that is the closest to the top of the list, ignoring all others. "Multiple" means that you can enter more than one line command simultaneously. See "Member Selection List Commands" on page 42 if you need more information about the line commands shown in the table.

Using Member Selection Lists

Primary Options	Type of Selection	Valid Line Command(s)	RENAME Field Available
Browse (1)	Single	S	No
Edit (2)	Single	S	No
Library (3.1)	Multiple	B,D,P,R	Yes
Move/Copy (3.3)	Multiple	S	Yes
Data Set List (3.4)	Multiple	B,D,E,P,R (see 1)	Yes
Reset (3.5)	Multiple	S	No
Convert (3.10)	Multiple	S	Yes
SuperC (3.12)	Multiple	S	No (see 2)
SuperCE (3.13)	Multiple	S	No (see 2)
Search-For (3.14)	Multiple	S	No
Foreground (4)	Single	S	No
Batch (5)	Single	S	No

Figure 11. Member Selection List Differences

The following items explain differences referred to in the preceding table:

1. B (browse data set), D (delete data set), E (edit data set), P (print data set), and R (rename data set) are valid only when the M (display member list) line command is used on a data set list. You can also enter TSO commands and CLISTs.

S (select) is valid also, but only when the B or E line commands are used on a data set list.
2. Instead of a RENAME field, this member list has an OLD-MEM field, which you can use to enter the name of a member in the “old” data set. See “SuperC Member Lists” on page 176 for more information about this field.

Displaying Member Lists

For each of the primary options listed in the preceding table, except Data Set List (3.4), you can display a member list by:

- Leaving the MEMBER field blank, for an ISPF library
- Omitting the member name from the name of an “other” partitioned data set.

- Entering a pattern as the member name.

A *pattern* is a partial member name followed by an asterisk. The pattern, including the asterisk, can contain no more than eight characters. For example, the pattern PROGRAM* might display the following list:

```
PROGRAM1  
PROGRAM2  
PROGRAM3  
PROGRAM4  
PROGRAM5
```

When using the Data Set List utility (3.4), you can display a member list by:

- Entering the M (display member list) line command
- Entering the E (edit) or B (browse) line command and then using one of the methods described in the preceding list. This applies only if you are editing or browsing members of a partitioned data set.

Ending Member Lists

To leave most member lists, enter the END command (PF3/15) or use the jump function (=) to go to another option. However, for SuperC and Search-For member lists, enter the RETURN (PF4/16) or CANCEL command, or the jump function. On these member lists, the END command processes your selections.

ISPF Member Statistics

On member lists, column headings are displayed in the national language. The information shown under the column headings contains the *ISPF statistics* that PDF has generated for each member. You can print these statistics using option X (print index listing) of the Library utility (3.1) or option P (print data set list) of the Data Set List utility (3.4). You can also use the SAVE command to write a member list or data set list to the ISPF list data set or to a sequential data set. The statistics are displayed next to each member name.

Figure 12 on page 40 shows an example of a member list with statistics and a one-character line command field to the left of the member names.

Using Member Selection Lists

MEMBER LIST -- FBROWN.PRIVATE.PANELS -----										ROW 0001 OF 00257
COMMAND ----> _										SCROLL ----> CSR
NAME	LIB	VV.MM	CREATED	CHANGED	SIZE	INIT	MOD	ID		
SRB00	3	21.11	82/10/26	83/12/05 09:12	29	23	0	DONLEY		
SRB01	2	20.04	82/10/26	85/12/13 08:51	29	23	0	MARTIN		
SRB10	3	21.00	84/03/26	84/03/26 10:53	29	29	0	PATTEN		
SRB15	3	20.03	82/10/26	83/12/05 09:12	26	23	0	BROWN		
SRB16	3	20.01	82/10/26	82/11/05 09:24	20	20	0	KISH		
SRB18	2	21.01	84/03/26	86/01/21 14:03	17	11	0	BERNACK		
SRB50	3	21.00	84/03/26	84/03/26 10:53	24	24	0	WALTERS		
SRC00	1		ALIAS							
SRC60	1	21.02	81/08/31	83/11/10 11:48	11	8	0	ROGERS		
SRE00	3	21.01	83/01/31	83/02/04 11:05	26	26	0	SPENCER		
SRE01	3	21.04	83/01/31	83/07/18 09:58	29	29	0	RICHEY		
SRE02	3	21.11	83/01/31	83/07/18 10:00	15	8	0	GURLEY		
SRE03	3	21.04	83/01/31	83/07/18 10:02	29	26	0	DUFF		
SRE04	3	20.16	83/01/31	83/06/22 11:03	29	44	0	COONS		
SRE05	2	01.01	85/06/18	85/10/04 11:13	29	29	0	RUPKAL		
SRE06	3	20.10	83/01/31	84/02/17 14:31	29	29	0	MANNINO		
SRE07	3	20.02	85/03/12	85/06/17 14:56	29	29	0	KELLERMA		
SRE08	2	20.01	83/01/31	86/01/20 09:04	23	20	0	MCINDOO		
SRE09	2	21.13	82/09/08	85/12/13 08:52	29	1	0	DARKSE		
SRE10	3	20.03	83/01/31	83/12/05 09:13	29	29	0	SEABOLD		

Figure 12. Member List Display

PDF generates statistics each time you edit a member, unless you enter the STATS OFF Edit primary command. The following fields identify the statistics in a member list:

NAME	Name of the member.
LIB	Library number. The LIB field is displayed only if you specify a concatenated sequence of libraries. It shows the library that contains the member. For example, if the member resides in the second library in the sequence, a "2" appears in the LIB field.
VV.MM	Version number and modification level. The version number is set to 1 and the modification level is set to 0 when the member is created. The modification level is the number of times this version has been modified. For example, "VV.MM 02.15" means version 2, modification 15. If a member name is just an alternate name for another member, ALIAS appears in this field.
CREATED	Date this version was created. The format used depends on your national format. For example, "85/06/27" means June 27, 1985, but so does "06/27/85" and "27/06/85."
CHANGED	Date and time this version was last modified; date is shown in the national format (see CREATED on page 40). Time is shown using a 24-hour format. For example, "17:20" means 5:20 p.m..
SIZE	Current number of lines. The largest number this field can display is 65,535.

Using Member Selection Lists

INIT	Number of lines when the member was first saved. The largest number this field can display is 65,535.
MOD	Number of lines in the current member that have been added or changed. If the data is unnumbered, this number is zero. The largest number this field can display is 65,535.
ID	The user ID of the person who created or last updated this version.

During Browse and Edit, the current version and modification level are displayed in the title area, line 1, following the library and member name. You can change the version number, the user ID, or both with the Reset ISPF Statistics utility (3.5) or with the LEVEL and VERSION Edit primary commands. Changing the version number updates most of the other statistics.

If you use the PDF editor to delete all the lines in a member of an ISPF library and then save the member, the statistics show that the member still exists but has a length of zero. The way to delete a member, including its statistics, is to use the Library utility (3.1).

Member lists displayed using the Data Set utility (option 3.4) contain an extended line command area and do not display the creation date. See Figure 60 on page 140 for an example.

Load Module Library Member Statistics

Figure 13 shows that the ISPF library statistics displayed in a member list have a different format for load module libraries.

MEMBER LIST -- FBROWN.PRIVATE.LOAD -----						ROW 00001	OF 00068
COMMAND ==> -						SCROLL	==> CSR
NAME	LIB	SIZE	TTR	ALIAS-OF	AC	ATTRIBUTES	
IGX00036	2	0010F0	000D06		00 FO	RN RU	
ISPF	2	000440	000313	ISRPCP	00 FO	RN RU	
ISRBRO	2	000BD8	000F09		00 FO	RN RU	
ISRECMBR	2	000138	000025		00 FO	RN RU	
ISREDIT	2	000EC8	000F23		00 FO	RN RU	
ISRFMT	2	0028F0	000A06		00 FO	RN RU	
ISRFPR	2	001290	000F01		00 FO	RN RU	
ISRFPT	2	000278	00002D		00 FO	RN RU	
ISRGE77	2	000768	000034		00 FO	RN RU	
ISRGE77A	2	000768	000106		00 FO	RN RU	
ISRGE78	2	000768	00010E		00 FO	RN RU	
ISRGE78A	2	000768	000116		00 FO	RN RU	
ISRJB1	2	000B40	00011E		00 FO	RN RU	
ISRJB2	2	0012F8	000F11	????????	?? FO	RN RU	
ISRKA78K	2	000768	000126		00 FO	RN RU	
ISRLALIN	2	007210	000A11		00 FO	RN RU	
ISRLCFCL	2	011B00	000738		00 FO	RN RU	
ISRLEMV	2	009048	000D0E		00 FO	RN RU	
ISRLEQDS	2	0006B8	00012E		00 FO	RN RU	
ISRLESS	2	000ED8	000136		00 FO	RN RU	
ISRLFMPP	2	0014D0	000B0A		00 FO	RN RU	

Figure 13. Load Module Library Display

Using Member Selection Lists

The fields on a member list display for a load module library are described below:

NAME	Name of the member.																		
LIB	Library number. The LIB field is displayed only if you specify a concatenated sequence of libraries. It shows the library that contains the member. For example, if the member resides in the second library in the sequence, a "2" appears in the LIB field.																		
SIZE	Size of the member. The largest number this field can display is 'FFFFFF'X.																		
TTR	Relative block address.																		
ALIAS-OF	Name of the member for which this member is an alias.																		
AC	Authorization code.																		
ATTRIBUTES	The member's attributes, as follows: <table><tr><td>FO</td><td>Process only by F-level linkage editor.</td></tr><tr><td>NX</td><td>Not executable.</td></tr><tr><td>OL</td><td>Can be loaded only.</td></tr><tr><td>OV</td><td>In overlay structure.</td></tr><tr><td>RF</td><td>Refreshable.</td></tr><tr><td>RN</td><td>Can be re-entered.</td></tr><tr><td>RU</td><td>Reusable.</td></tr><tr><td>SC</td><td>Scatter format.</td></tr><tr><td>TS</td><td>Module to be tested.</td></tr></table>	FO	Process only by F-level linkage editor.	NX	Not executable.	OL	Can be loaded only.	OV	In overlay structure.	RF	Refreshable.	RN	Can be re-entered.	RU	Reusable.	SC	Scatter format.	TS	Module to be tested.
FO	Process only by F-level linkage editor.																		
NX	Not executable.																		
OL	Can be loaded only.																		
OV	In overlay structure.																		
RF	Refreshable.																		
RN	Can be re-entered.																		
RU	Reusable.																		
SC	Scatter format.																		
TS	Module to be tested.																		

Question marks (?) are placed in the member list of a load module library for members that have invalid load module directory fields. For example, member ISRJB2, shown in Figure 13 on page 41, contains invalid alias and authorization code information.

Load module library lists displayed using the Data Set utility (option 3.4) contain an extended line command area and do not display the creation date. See Figure 61 on page 141 for an example.

Member Selection List Commands

If the member list is too large for the screen, you can see other parts of the list by using the UP and DOWN scroll commands. These commands are valid for all member list displays. However, since a member list display can be no wider than 80 characters, the LEFT and RIGHT scroll commands are inoperative. See "Scrolling Within a Data Display" on page 25 for more information about scroll commands.

The following *primary* commands are valid for all member list displays. You enter these commands in the COMMAND field:

- LOCATE string
- RESET
- SAVE [list-id]
- SELECT pattern [lcmd]
- SORT [field1[field2]]

The following *line* commands can be used with member lists. These are one-character commands that are entered to the left of the member name. The option you are using determines:

- Whether you can enter more than one line command simultaneously
- Which line commands are valid
- Whether a RENAME or OLD-MEM field is available.

Figure 11 on page 38 provides a quick reference to the differences between member lists and the line commands available on each one. The line commands are:

- S (select member)

The S line command is available for all member list displays *except* the Library and Data Set List utilities. See “S Line Command” on page 45 for more information.

- B (browse member)
- D (delete member)
- E (edit member)
- P (print member)
- R (rename member)

The B, D, P, and R member list line commands are available *only* for the Library and Data Set List utilities. The E line command is available only for the Data Set List utility. However, for the Data Set List utility, these line commands are valid only after you enter the M (Display member list) line command. See “Library and Data Set List Utility Line Commands” on page 47 for information.

PDF ignores any unprocessed member list commands when you leave a member list.

Locating a Data String

To find a data string, you can enter a LOCATE command in the COMMAND field on any member list display. The format of the command is:

LOCATE string

where:

string A data string that is used to find an entry based on how the member list is sorted.

Using Member Selection Lists

PDF searches the field by which the member list was sorted for an entry equal to "string." Either the entry, if found, or else the entry that immediately precedes the entry that you were searching for is scrolled to the top of the list.

For example, the member list shown in Figure 13 on page 41 is sorted by name. Therefore, the following command causes member ISRJB1 to scroll to the top of the list:

Removing Unwanted Line Commands and Messages

The RESET command removes unprocessed line commands and messages that show the result of line command processing. This command has no operands.

Writing a Member List to a Sequential Data Set

The SAVE primary command writes a member selection list to a sequential data set. The format of the SAVE command is:

```
SAVE [list-id]
```

where:

list-id The name of a sequential data set. PDF names the data set:

```
prefix.userid.list-id.MEMBERS
```

where:

prefix Your TSO user prefix if you have one and if it is different from your user ID.

userid Your TSO user ID.

The data set is created if it does not exist, or written over if it exists, and has compatible attributes.

PDF writes the member list in the current sort order and as it appears on the display, except for the column headings, line command fields, and anything you have typed on the display.

If you omit the list-id, PDF writes the member selection list in the current sort order to the ISPF list data set. Processing is the same as using option X of the Library utility (primary option 3.1), except that data set information is not printed.

Selecting a Member

The SELECT, or S, command can be used as either a primary command or a line command.

SELECT Primary Command:

The SELECT primary command allows you to select one or more members in a member list, whether they are displayed or not. When entered on a member list displayed using the Edit option, it even creates a member if you specify the complete member name of a member that does not exist.

The SELECT command optionally provides a quick method of invoking the same line command for one or more members. The format of the SELECT command is:

```
SELECT {pattern | * }[lcmd]
```

where:

- pattern** Either a complete member name or a partial member name followed by an asterisk. See "Displaying Member Lists" on page 38 if you need more information about using patterns.
- *** An asterisk, which means you want to select *all* members in a member list.
- lcmd** One of the following optional line commands: S (select), B (browse), D (delete), E (edit), or P (print). On a member list that has an expanded line command area, such as one generated by the M (member list) line command in the Data Set List utility (3.4), you can also enter a TSO command or CLIST. If you do not enter a line command, S is the default.

The member list shown in Figure 12 on page 40 contains members ISRC00 and ISRC60. The following command selects these members for printing:

```
SELECT ISRC00 ISRC60 P
```

S Line Command:

You can enter the S line command at the beginning of a line, ahead of one or more member name(s). For example, in Figure 13 on page 41, you could select member ISRFPT by moving the cursor to the left of the member name, typing S, and pressing the ENTER key.

Note: On member lists displayed with the Browse, Edit, Foreground, and Batch options, PDF processes only the first S entered, ignoring all others.

With the Move/Copy and Convert utilities, you can rename members by entering new member names in the RENAME field to the right of the member name. However, to rename when promoting it, you must have authority to promote the member under the new name. A member with the old name in the "to" library is neither replaced nor freed.

Using Member Selection Lists

Sorting a Member List

The SORT primary command arranges a member list according to the fields you specify. The sort sequence, ascending or descending, is determined by the field(s) you choose and is maintained between member list displays. The format of this command is:

```
SORT [field1[field2]]
```

where:

field1 The primary field by which the member list is sorted.

field2 The secondary field by which the member list is sorted.

The following tables show:

- Valid values for field1 and field2
- The sort sequence used for each field
- A description of each field name.

Field	Sequence	Description
NAME	Ascending	Member name
LIB	Ascending	Library in concatenation sequence
VV	Ascending	PDF version number
MM	Ascending	PDF modification level
CREATED	Descending	Creation date
CHANGED	Descending	Date and time last changed
SIZE	Descending	Current number of records
INIT	Descending	Initial number of records
MOD	Descending	Number of modified records
ID	Ascending	Last user

Figure 14. SORT Fields for Source Libraries

Field	Sequence	Description
NAME	Ascending	Member name
LIB	Ascending	Library in concatenation sequence
SIZE	Descending	Load module size
TTR	Ascending	TTRN of beginning of load module
ALIAS-OF	Ascending	Member this is an alias of
AC	Ascending	Authorization code
ATTRIBUTES	Descending	Load module attributes

Figure 15. SORT Fields for Load Libraries

For example, to sort a member list by size and then by track record, you could enter the following:

Library and Data Set List Utility Line Commands

On member list displays for the Library utility (3.1) and the Data Set List utility (3.4), you can enter the following line commands at the beginning of a line, ahead of one or more member name(s).

B	Browse the member.
D	Delete the member.
E	Edit the member.
P	Print the member.
R	Rename the member.

Notes:

1. *In the Data Set List utility, these line commands are valid when you enter the M (display member list) line command to display a list of data set members. However, for member lists displayed using the E (edit data set) and B (browse data set) line commands, only the S (select) line command is valid.*
2. *Member lists displayed with the M line command have a nine-character line command field to accommodate TSO commands and CLISTs. The E (edit data set member) line command, shown in the preceding list, is valid only with these member lists. It is not valid for member lists displayed by the Library utility. See "M - Display Member List" on page 139 and "TSO Commands and CLISTs" on page 132 for more information.*

When you use the R line command, enter the new member name in the RENAME field to the right of the member name.

When you press the ENTER key, each member preceded by a line command is processed unless:

- A B (browse) or E (edit) line command is followed by another line command. When you return to the member list after browsing or editing a member, you must press the ENTER key again to invoke any remaining line commands.
- You enter a line command for a member that was deleted. The names of deleted members are not removed from the member list until it is updated. Remove the line command that precedes the deleted member and press the ENTER key again. See "Updating a Member List" on page 49 for more information.
- An R (rename) line command is entered, but you fail to put a new name in the RENAME field. Enter a new member name and press the ENTER key again.

You can then do one of the following:

- Enter additional primary or line commands
- Scroll, if necessary, to bring additional members into view
- Enter the END command to return to the previous panel.

Using Member Selection Lists

Figure 16 on page 48 shows a before-and-after example that prints members ACCT1 and ACCT2, deletes member UPDATE, and renames member LISTNEW to LISTOUT.

LIBRARY - ISPFDEMO.MYLIB.PLI -----									
COMMAND ---->									
	NAME	RENAME	VV.MM	CREATED	CHANGED	SIZE	INIT	MOD	ID
	ACCOUNT		01.00	85/01/09	85/01/09 17:07	21	21	0	BERNACK
P	ACCT1		01.01	85/02/11	85/04/23 14:52	199	193	0	BROWN
P	ACCT2		01.00	85/03/09	85/03/09 17:07	20	20	0	MARTIN
	COINS		01.04	85/04/24	85/04/28 16:20	19	19	4	PATTEN
	COMPX		01.00	85/01/21	85/01/21 11:08	44	44	0	ROGERS
	COMPY		01.01	85/01/14	85/01/16 12:30	13	13	1	RICHEY
	DCLS		01.00	85/04/23	85/04/23 15:14	20	20	0	KISH
R	LISTNEW	LISTOUT	01.02	85/04/23	85/05/06 10:00	17	13	6	SPENCER
	MAIN		01.00	85/01/09	85/01/09 17:08	4	4	0	WALTERS
D	TESTDIR		01.02	85/04/23	85/05/06 09:04	30	43	10	DONLEY
END									

LIBRARY - ISPFDEMO.MYLIB.PLI -----									
COMMAND ---->									
	NAME	RENAME	VV.MM	CREATED	CHANGED	SIZE	INIT	MOD	ID
	ACCOUNT		01.00	85/01/09	85/01/09 17:07	21	21	0	BERNACK
	ACCT1	*PRINTED	01.01	85/02/11	85/04/23 14:52	199	193	0	BROWN
	ACCT2	*PRINTED	01.00	85/03/09	85/03/09 17:07	20	20	0	MARTIN
	COINS		01.04	85/04/24	85/04/28 16:20	19	19	4	PATTEN
	COMPX		01.00	85/01/21	85/01/21 11:08	44	44	0	ROGERS
	COMPY		01.01	85/01/14	85/01/16 12:30	13	13	1	RICHEY
	DCLS		01.00	85/04/23	85/04/23 15:14	20	20	0	KISH
	LISTNEW	*RENAMED							
	MAIN		01.00	85/01/09	85/01/09 17:08	4	4	0	WALTERS
	TESTDIR	*DELETED							
END									

Figure 16. Library Utility - Before and After Print, Rename, and Delete

Updating a Member List

When the member list is displayed again following completion of a function or command, it does *not* include:

- For Browse and Edit:
 - New members created by a recursive invocation of Edit
 - Members created by another user
 - In split-screen mode, members created on another logical screen.
- For Library, Move/Copy, Data Set List, and Convert:
 - New names of members that have been renamed
 - Fewer member names when members are deleted (Library and Data Set List) or moved (Move/Copy).

To display an up-to-date list, return to the previous panel, leave the member name blank or enter a pattern, and press the ENTER key. See “Displaying Member Lists” on page 38 for more information about displaying member lists.

Using Data Set Passwords

A DATA SET PASSWORD field is included on library/data set entry panels:

```
DATA SET PASSWORD ==>
```

where:

DATA SET PASSWORD

The password for OS password-protected data sets. By assigning more than one password to the same data set, you can give some users read-only access while giving others read-write access.

Non-display input fields are used so that the passwords do not appear on the screen. When you specify a concatenated sequence of libraries, the password applies to all data sets in the sequence.

If you replace a long password with a shorter password, blank out the remaining spaces of the password field.

PDF can be used with either the TSO/VS2 Programming Control Facility (PCF) or the Resource Access Control Facility (RACF). PCF and RACF provide extensive facilities for data set security. However, when using either PCF or RACF, do not try to enter a password on the PDF panels, since both of these facilities rely on your TSO user ID and logon password to identify you and check for proper authorization.

Using Mixed Mode

Using Format Definitions

A **FORMAT NAME** field is included on the Browse - Entry Panel and on the Edit - Entry Panel:

FORMAT NAME ===>

where:

FORMAT NAME

The name of a format definition, which is used to browse or edit a *formatted* data set. A formatted data set contains records that consist of subfields. The locations and lengths of these subfields are fixed throughout the data set. The formatted data Edit and Browse are particularly useful when a field contains Double-byte Character Set (DBCS) data without shift-out (SO) and shift-in (SI) characters.

The format name can consist of up to eight alphanumeric characters, the first of which must be alphabetic.

A format definition can include Extended Binary Coded Decimal Interchange Code (EBCDIC) fields, DBCS fields, and mixed fields. If the specified format includes a mixed field definition, the **MIXED MODE** field is ignored, even if you specify a value of NO. See "Using Mixed Mode" for information about the **MIXED MODE** field.

For information about defining formats for formatted data sets, see "Format Specification Utility (Option 3.11)" on page 166. The Format Specification utility is provided to support the IBM 5550 terminal using DBCS.

When formatted data is displayed, an attribute character that does not reside in the data set and is not stored in the data set precedes each field. Therefore, the column position on the display is different from the column position in the data set.

The allowable maximum length is decreased two bytes per field definition from the standard Browse and Edit allowable maximum length.

Using Mixed Mode

A **MIXED MODE** field is included on the Browse - Entry Panel and on the Edit - Entry Panel:

MIXED MODE ===> (YES or NO)

where:

MIXED MODE

Shows whether you want to browse or edit unformatted mixed data that contains both EBCDIC (single-byte) and DBCS (double-byte) characters.

Valid values for the MIXED MODE field are:

- | | |
|------------|---|
| YES | To browse or edit data that contains both EBCDIC and DBCS characters. |
| NO | To browse or edit data that contains EBCDIC characters only. This is the default value. |

If your terminal does not support DBCS, the value in this field is ignored.

DBCS strings are enclosed with SO (X'0E') and SI (X'0F') characters in unformatted mixed data. The SO character precedes the DBCS character string and the SI character follows the string.

If the browse or edit line contains invalid mixed data, PDF assumes the line can contain only EBCDIC characters. Examples of invalid mixed data include:

- Unpaired SO and SI characters
- Invalid DBCS characters between SO and SI characters
- An odd number of bytes between SO and SI characters.

If you invoke Browse or Edit from the the Data Set List utility (3.4), or Browse from the Library utility (3.1), PDF assumes you want to use mixed mode.

If you want to browse or edit DBCS data as EBCDIC data, you must browse or edit in non-mixed mode. You can do this by operating from a terminal that does not support DBCS or by specifying **NO** in the MIXED MODE field.

In non-mixed mode, SO and SI characters are not treated as special characters; instead, they are treated as characters that cannot be displayed. Thus, you can browse or edit the data in the conventional way.

You can also browse or edit DBCS data in hexadecimal format, just as you would EBCDIC data. For information about specifying hexadecimal display, see "HEX - Displaying Data in Hexadecimal Format" on page 95. You should *not* edit a record in hexadecimal format, however, when a DBCS string encroaches on the display boundary. If you edit under this condition, you may get unwanted results.

Invalid DBCS data is not supported. If DBCS fields or DBCS strings in a mixed field contain any bytes with hexadecimal code ranging from X'00' to X'3F', you may get unwanted results.

Using Packed Data Sets

The *packed* data format lets you use direct access storage devices (DASDs) more efficiently. In this format, PDF replaces any repeating characters with a sequence showing how many times the character is repeated. Before you can properly use data stored in this format as input to processing

Using List and Log Data Sets

programs, such as compilers, you must first tell PDF to unpack and expand the data.

The two requirements for using packed data sets are:

- To store data in packed format:
 - Enter the PACK ON Edit primary command while editing a data set or PDS member.
 - Enter YES in the SPECIFY PACK OPTION FOR “TO” DATA SET field when copying or moving members, or in the PACK DATA field when promoting members by using the Move/Copy utility (3.3).
- To unpack and expand packed data for processing, enter YES in the SOURCE DATA PACKED field on the Foreground Selection Panel or the Batch Selection Panel. You must enter YES in this field if *any* of the input data, including that referred to in COPY or INCLUDE statements, is in packed format.

Using List and Log Data Sets

PDF helps you get hardcopy listings of source modules, and also maintains a log of significant user activities. These items are kept in data sets called the list data set and the log data set, respectively.

When needed, the two data sets are allocated automatically. They are temporary data sets named:

```
prefix.user-id.SPFn.LIST  
prefix.user-id.SPFLOGn.LIST
```

where:

prefix The data set prefix in your TSO profile. It is used only if you have one and it is different from your user ID.

userid Your user ID.

n A number from 0 to 9.

If you have specified in your TSO profile a data set prefix that differs from your user ID, the data set names begin with your data set prefix, followed by your user ID. Once generated, these data sets remain open throughout your PDF session.

CHAPTER 5. EDIT (Option 2)

This chapter provides a general description of Edit, option 2 on the ISPF/PDF Primary Option Menu, which allows you to create, display, and modify source data such as program code, test data, and documentation.

For complete information, see *Edit and Edit Macros*.

List Data Set

The list data set is generated the first time you request a print function. It is used to collect PDF-produced listings and screen “snapshots.”

The list data set accumulates, at your option, a listing of any source module created or changed by the PDF editor. You can also use the PDF utilities to get source listings and other types of printed output.

You can get a screen “snapshot” by entering the PRINT, PRINT-HI, PRINTL, or PRINTLHI command. You can also get listings of member selection lists and data set lists by using the SAVE command.

To avoid generating an ISPF list data set, do not request any print functions.

Log Data Set

The log data set is generated the first time you do some action that results in a log message, such as saving edited data or submitting a Batch job. It maintains a log of your significant activities, including ISPF log messages, a record of data sets and members you changed using the Edit or Utilities options, Batch jobs you submitted, Dialog Test trace data, and so forth.

Use the Log/List option (0.2) to prevent generation of the ISPF log data set. However, if you use the Dialog Test option (7), you should allow generation of the log data set, because Dialog Test writes trace data to the log when you request it. Also, if Dialog Test finds an unexpected condition, problem data and error messages are written to the log.

Printing List/Log Data Sets

Appendix A, “Listing Formats” shows and describes the format of the ISPF list and log data sets. You can print these data sets when you end ISPF by directing the output to:

- A system output device via a Batch job
- A local IBM 3284, 3286, 3287, 3288, or 3289 printer.

If the list and log data sets are printed, they are automatically deleted following printing; new list and log data sets are allocated the next time you invoke PDF. If you decide not to print the log or list data sets, you can either:

- Delete them
- Keep them and allocate them again in the next session
- Keep them, but allocate a new data set in the next session.

Other Temporary Data Sets

Foreground and Batch Output Listings

The following additional listing data sets are allocated as needed for Foreground or Batch processing:

```
prefix.userid.list-id.LIST  
prefix.userid.list-id.LINKLIST  
prefix.userid.list-id.TERM  
prefix.userid.list-id.TESTLIST
```

where:

- prefix** The data set prefix in your TSO profile. It is used only if you have one and it is different from your user ID.
- userid** Your user ID.
- list-id** The name specified in the LIST ID field on the Foreground or Batch data entry panel. This name is required for sequential data sets. However, for partitioned data sets, the member name becomes the default list-id if the LIST ID field is blank.

The particular data set names used depend on the Foreground or Batch processing option chosen.

For Batch processing, the output can either be directed to a list data set or printed as part of the Batch job. When Batch processing is finished, you can browse the list data set and then invoke the Hardcopy utility (3.6) to print it. Using this utility, show whether you want to keep the data set or delete it after printing. PDF does not delete these data sets when you end ISPF.

For the Foreground option, the output listing is directed to a list data set and automatically displayed for browsing. When you end the browse function, PDF displays a selection panel that lets you choose whether to print, keep, or delete the list data set. Again, PDF does not delete this data set when you end ISPF.

Other Temporary Data Sets

If you are using virtual I/O (VIO), you can allocate space for temporary data sets, to which VIO assigns system-generated names. Otherwise, PDF allocates temporary control and listing data sets, as needed, for its own internal use. They are assigned the following names:

```
prefix.user-id.SPFTEMPn.CNTL  
prefix.user-id.SPFTEMPn.LIST  
prefix.user-id.appl-idzzzz.BACKUP  
prefix.user-id.appl-idzzzz.BACKUPI  
prefix.user-id.SPFxxx.OUTLIST
```

Job Statement Information

The following parameters are used by temporary data sets:

- prefix** The data set prefix in your TSO profile. It is used only if you have one and it is different from your user ID.
- userid** Your user ID.
- n** A number from 1 to 4, depending on which logical screen is active.
- appl-id** The application ID.
- zzzz** A number from 0001 to 0008, or higher if customized, controlled by the edit recovery table (appl-id EDRT for the EDREC service and appl-id EIRT for the EDIREC service) and the number of concurrent Edit invocations active.
- xxx** A number generated by PDF.

These data sets are deleted:

- By edit recovery when no longer needed
- When you specifically request that they be deleted.

You are usually not aware of their existence.

Job Statement Information

PDF lets you submit Batch jobs for printing and language processing. However, before submitting a Batch job, you must supply job statement information. For this purpose, four lines are provided on each job submission panel.

The first time a job submission panel is displayed, the job statement information appears as follows:

```
JOB STATEMENT INFORMATION: (Required for system printer)
===> //user-ida JOB (ACCOUNT),'NAME'
===> //*
===> //*
===> /*
```

These are the JOB STATEMENT INFORMATION parameters:

- user-ida** The pre-entered job name, which consists of your TSO user ID with the letter "A" as the last character. This character changes to "B," then "C," and so forth, each time the job statement information is used. If you prefer, you can change the last character of the job name to a number instead of a letter. This causes the last character to cycle from 0 to 9, instead of from A to Z.

If you type over the complete job name, automatic incrementing of the last character stops, unless the job name starts with your user ID.

Running a Sample PDF Session

Note: If you want to query TSO regarding your job status, do not change the pre-entered user ID or remove the character at the end. These actions prevent TSO from finding the job when you ask for status.

You must enter the remaining job statement information the first time you submit a Batch job. PDF prompts you by displaying the following JOB parameters:

- | | |
|------------------|---|
| (ACCOUNT) | The (ACCOUNT) information varies from one installation to another, depending on the TSO version being used. It is used to make sure you receive the results of your job and must be enclosed in parentheses. Typical examples of account information are your employee serial number and the box number where your printouts are delivered. |
| 'NAME' | The 'NAME' information is a text string that must be enclosed in single quotation marks. It can consist of your name, your telephone number, or any other information you care to provide. |

You can use the lines containing “/*”:

- As continuation lines by removing the asterisk (*)
- To enter other JCL statements, such as JOBLIB DD.

If you do not need these lines, you can blank them out. Blank lines are not submitted to the job stream.

Running a Sample PDF Session

This sample PDF session is an example of PDF terminal operations. For new users, it is a quick introduction to PDF. For users with previous PDF experience, it is a quick review. It can also be useful as a demonstration that PDF has been properly installed and is operational.

The scenario requires the installation of a data set named SYS1.SAMPLIB. This data set is included on the PDF basic distribution tape and should contain the following four members:

- | | |
|-----------------|--------------------------|
| ISRASM | Sample Assembler Source. |
| ISRCOBOL | Sample COBOL Source. |
| ISRFORT | Sample FORTRAN Source. |
| ISRPLI | Sample PL/I Source. |

During the scenario, member ISRASM is copied from SYS1.SAMPLIB to a user data set. The other three members are not used during the scenario.

The scenario also assumes the TSO Assembler Prompter, program number 5734-CP2, is installed and available. If the Assembler Prompter is not available at your installation, skip steps 35 through 48.

Action	Result
1. Log on to TSO.	A Ready message appears on your screen.
2. Type PDF (or the appropriate alias) and press the ENTER key.	The ISPF/PDF Primary Option Menu is displayed.
3. On the OPTION line, type 3 to select the Utilities option. Then press the ENTER key.	The Utility Selection Menu is displayed.
4. On the OPTION line, type 2 to select the Data Set utility. Then press the ENTER key.	The Data Set Utility panel is displayed.
5. On the OPTION line, type A to allocate a new data set. Specify an ISPF library by typing the following information, but substitute your first name in the GROUP field: PROJECT ===> GROUP ===> TYPE ===> Now press the ENTER key.	The Allocate New Data Set panel is displayed.
6. Type the following information: <i>Note: Leave the VOLUME SERIAL, GENERIC UNIT, and EXPIRATION DATE fields blank.</i> SPACE UNITS ===> PRIMARY QUANTITY ===> SECONDARY QUANTITY ===> DIRECTORY BLOCKS ===> RECORD FORMAT ===> RECORD LENGTH ===> BLOCK SIZE ===> Now press the ENTER key.	Data set SAMPLE.name.ASM is allocated on scratch volume. The Data Set Utility panel is displayed again.
7. Select option A again and specify the following ISPF library, again substituting your first name in the GROUP field: PROJECT ===> GROUP ===> TYPE ===> Press the ENTER key.	The Allocate New Data Set panel is displayed again.
8. Leave everything the same. Just press the ENTER key.	Data set SAMPLE.name.OBJ is allocated. The Data Set Utility panel is displayed again.
9. Press PF3 , the END command.	The Utility Selection Menu is displayed again.
10. Press PF3 again.	The ISPF/PDF Primary Option Menu is displayed again.
11. Type 3.3 to select the Move/Copy utility, bypassing the Utility Selection Menu. Press the ENTER key.	The Move/Copy Utility panel is displayed.

Running a Sample PDF Session

Action	Result
<p>12. Now you will copy a data set. On the OPTION line, type C to select Copy data set or member(s). Then, under FROM OTHER PARTITIONED OR SEQUENTIAL DATA SET:, enter the following data set name:</p> <p>DATA SET NAME ===> SYS1.SAMPLIB(ISRASM)</p> <p>Press the ENTER key.</p>	<p>A panel titled COPY --- FROM SYS1.SAMPLIB(ISRASM) --- is displayed.</p>
<p>13. Under TO ISPF LIBRARY:, type the following values:</p> <p>PROJECT ===> SAMPLE GROUP ===> NAME TYPE ===> ASM</p> <p>Press the ENTER key.</p>	<p>Member ISRASM is copied from data set SYS1.SAMPLIB to ISPF library SAMPLE.name.ASM. Then, the Move/Copy Utility panel is displayed again.</p>
14. Press PF3 .	The ISPF/PDF Primary Option Menu is displayed.
15. Now you will browse member ISRASM . On the OPTION line, type 1 to select Browse and press the ENTER key.	The Browse - Entry Panel is displayed.
<p>16. Type the following under ISPF LIBRARY:</p> <p>PROJECT ===> SAMPLE GROUP ===> NAME TYPE ===> ASM</p> <p><i>Note: Leave the MEMBER field blank.</i></p> <p>Press the ENTER key.</p>	<p>A member list is displayed, showing ISRASM as the only member in the SAMPLE.name.ASM library.</p>
17. Move the cursor to the left of ISRASM . Then type S to select ISRASM and press the ENTER key.	A panel titled BROWSE - SAMPLE.name.ASM (ISRASM) is displayed. This is the first page of member ISRASM .
18. Press PF8 to scroll ahead one page.	The second page of ISRASM is displayed.
19. Press PF7 to scroll backward one page.	The first page of ISRASM is displayed again.
20. Type FIND COMMENT on the COMMAND line and press the ENTER key.	The cursor moves to the first occurrence of the character string COMMENT and the string is intensified. Also, the message CHARS 'COMMENT' FOUND is displayed in the upper-right corner of the screen.
21. To find the next occurrence of COMMENT , press PF5 , the RFIND command.	The cursor moves to the second occurrence of COMMENT and once again the string is intensified.
22. Press PF3 .	The member list is displayed again.
23. Press PF3 again.	The Browse - Entry Panel is displayed again.
24. Press PF3 one more time.	The ISPF/PDF Primary Option Menu is displayed.

Action	Result
25. Now you will edit member ISRASM. On the OPTION line, type 2 to select Edit and press the ENTER key.	The Edit - Entry Panel is displayed.
26. Type ISRASM in the MEMBER field and press the ENTER key.	A panel titled EDIT --- SAMPLE.name.ASM(ISRASM) is displayed. This is the first page of member ISRASM.
27. On the COMMAND line, type FIND COMMENT and press the ENTER key to find the line containing the character string "COMMENT."	The cursor moves to the first occurrence of the character string and the line number is intensified.
28. Delete COMMENT by pressing the Erase EOF key.	COMMENT is erased. Any characters to the right of COMMENT would have been erased, also.
29. Press PF5 to find the next occurrence of COMMENT .	The cursor moves to the second occurrence of COMMENT and the line number is intensified.
30. Move the cursor to the sequence number of the line below COMMENT , then move the cursor up one line. Repeat the COMMENT line by typing R over the first digit of the line number and pressing the ENTER key.	The line is repeated.
31. On the COMMAND line, type AUTOLIST ON , then press the ENTER key.	Sets autolist mode on for automatic source listings.
32. Try out more Edit commands if you like, but remember: this program will be assembled later.	HAVE FUN!
33. Press PF3.	Member ISRASM is saved in data set SAMPLE.name.ASM and a listing of the member is placed in the ISPF list data set. The Edit - Entry Panel is displayed.
34. Press PF3 again.	The ISPF/PDF Primary Option Menu is displayed.
35. Now you will assemble the program. On the OPTION line, type 4 and press the ENTER key to select the Foreground option.	The Foreground Selection Panel is displayed.
36. On the OPTION line, type 1 and press the ENTER key to select Assembler H .	The Foreground Assembler H panel is displayed.
37. Type the following values: PROJECT ==> SAMPLE GROUP ==> DADS TYPE ==> ASM MEMBER ==> ISRASM LIST ID ==> ASCXEST. ASSEMBLER OPTIONS: ==> LIST,TEST,VENT Press the ENTER key.	The Assembler prompter is invoked. Terminal output is written at the bottom of a blank screen. When *** is displayed, press the ENTER key to continue. <i>Note: The assembly may take a few minutes.</i> The assembly listing is displayed in Browse mode.
38. Browse the listing using the scroll commands. Then press PF3.	The Foreground Print Options panel is displayed.

Running a Sample PDF Session

Action	Result
39. On the OPTION line, type K to select Keep data set (without printing) and press the ENTER key.	The assembled program is saved in the list data set and the Foreground Assembler H panel is displayed again.
40. Press PF3.	The Foreground Selection Panel is displayed again.
41. Press PF3 again.	The ISPF/PDF Primary Option Menu is displayed.
42. On the OPTION line, type 5 to select the Batch option and press the ENTER key.	The Batch Selection Panel is displayed.
43. On the OPTION line, type 1 to select Assembler H , enter job statement information as required by your installation, and press the ENTER key.	The Batch Assembler H panel is displayed.
44. Type the following values: PROJECT ===> GROUP ===> TYPE ===> MEMBER ===> LISTID ===> ASSEMBLER OPTIONS: ===> Press the ENTER key.	JCL for assembling ISRASM is generated. The Batch Selection Panel is displayed again.
45. Press PF3.	The job is submitted. The jobname is displayed at the bottom of the screen. When *** is displayed, press the ENTER key. The ISPF/PDF Primary Option Menu is displayed.
46. On the OPTION line, type 6 to select the Command option. Press the ENTER key.	The TSO Command Processor panel is displayed.
47. Type the following command: COMMAND ===> Press the ENTER key.	The status of your job is displayed. When *** is displayed, press the ENTER key. The TSO Command Processor panel is displayed again with the previous command still showing.
48. Press PF3.	The ISPF/PDF Primary Option Menu is displayed again.
49. On the OPTION line, type T to select the Tutorial option.	The beginning of the ISPF/PDF tutorial is displayed. Follow the directions to learn more about ISPF/PDF. When you have finished, press PF3 to return to the ISPF/PDF Primary Option Menu.

If you have followed this scenario, you should have an Assembler listing in a data set named either prefix.userid.ASMTEST1.LIST or userid.ASMTEST1.LIST. Also, when the job that you submitted has finished, you will have another listing in a data set named either prefix.userid.ASMTEST2.LIST or userid.ASMTEST2.LIST. Try the Browse option to review the listings and try experimenting with other ISPF/PDF options. When you have finished, return to the ISPF/PDF Primary Option Menu.

Action	Result
50. With the ISPF/PDF Primary Option Menu on the screen, press PF3.	The ISPF end panel is displayed.
51. Type PD in the Process option field for both the log and list data sets and fill in the job statement information as required by your installation. Then press the ENTER key.	A job is submitted to print and delete the data sets. The jobname is displayed at the bottom of the screen.
52. You are now out of ISPF/PDF. To leave TSO, type LOGOFF and press the ENTER key.	This is the end of this usage scenario.

CHAPTER 3. ISPF PARMS (Option 0)

This chapter describes ISPF Parms, option 0 on the ISPF/PDF Primary Option Menu.

The ISPF Parms option allows you to specify selected ISPF parameters to customize the terminal and your output listing formats. The parameters you can specify include:

- **Terminal characteristics**
- **ISPF console, log, and list defaults**
- **PF key definitions**
- **Screen display characteristics**
- **List data set characteristics**
- **PRINTG (print graphics) command parameters**
- **ENVIRON (environment) command parameters.**

This chapter describes the procedures to follow to specify these parameters.

Chapter 3. ISPF Parms (Option 0)

The ISPF Parms option (0) allows you to display and change selected ISPF parameters at any time during the PDF session. Changes remain in effect until you change the parameters again and are saved from session to session.

```
----- ISPF PARAMETER OPTIONS -----
OPTION  ===> _

1  TERMINAL  - Specify terminal characteristics
2  LOG/LIST  - Specify ISPF log and list defaults
3  PF KEYS   - Specify PF keys for 3278 terminal with 24 PF keys
4  DISPLAY   - Specify screen display characteristics
5  LIST      - Specify list data set characteristics
6  GRAPHIC   - Specify GDDM graphic print parameters
7  ENVIRON   - Specify ENVIRON command settings
```

Figure 17. ISPF Parameter Options Panel

The following sections describe the options shown at the top of the ISPF Parameter Options panel.

Terminal (Option 0.1)

When you select this option, the Terminal Characteristics panel is displayed. This panel allows you to specify the terminal type, the number of program function (PF) keys, the default pad character for panel input fields, the mode of operation for a 3278 Model 5 or 3290, and the command-stacking delimiter. The panel, with the first user defaults specified, is shown in Figure 18. The allowable alternatives for these defaults are indicated on the display.

Note: The information on this panel may vary, depending on the terminals that are supported for your country.

```
----- TERMINAL CHARACTERISTICS -----
COMMAND ----> _

TERMINAL TYPE      ----> 3278      (3277 - 3275/3277 terminal)
                                   (3277A - 3275/3277 with APL keyboard)
                                   (3278 - 3276/3278/3279/3290 terminal)
                                   (3278A - 3276/3278/3279 APL keyboard)
                                   (3278T - 3276/3278/3279 TEXT keyboard)
                                   (3290A - 3290 with APL keyboard)

NUMBER OF PF KEYS  ----> 24      (12 or 24)

INPUT FIELD PAD    ----> N      (N - Nulls) (B - Blanks) (Special Character-
                                   must not be the same as COMMAND DELIMITER)

COMMAND DELIMITER  ----> ;      (Special character for command stacking)

SCREEN FORMAT      ----> DATA   (Select one of the following:)
  (3278 Model 5 only)              (DATA - Format based on data width)
                                   (STD - Always format 24 lines by 80 chars)
                                   (MAX - Always format 27 lines by 132 chars)

                                   (3290 Only) (PART - Format using hardware partitions.
                                   Effective the next ISPF invocation.)
```

Figure 18. Terminal Characteristics Panel

After you review these parameters and change them as necessary, enter the END command to return to the previous panel. The fields on the Terminal Characteristics panel are:

TERMINAL TYPE

Specifying the terminal type allows ISPF to recognize the characters that your terminal can display. A 3278 terminal can display six more characters than a 3277. However, ISPF cannot sense the terminal type if you switch between a 3277 and 3278 Model 2, both of which are 24-line terminals, unless you show the change in this field. If you have a 3279 terminal, specify 3278 as the terminal type, since a 3279 terminal has the same character set as a 3278. See Appendix B, "Character Translations for APL and TEXT Keyboards" on page 363 for the APL and TEXT character set descriptions.

One or more of the following installation-dependent options for terminal type may be included on the Terminal Characteristics panel:

3278CF	For 3278 Canadian French terminals.
3277KN	For 3277 Katakana terminals.
3278KN	For 3278 Katakana terminals.

A 5550-3270 Kanji emulation Version 3 terminal has the same character set as a 3278 Katakana terminal. If you are using a 5550-3270 Kanji emulation Version 3 terminal, you should specify 3278KN as the terminal type.

To avoid problems in finding and changing text strings on a Katakana terminal while caps mode is off when the data was entered using caps mode on, set the terminal type to 3278KN.

NUMBER OF PF KEYS

Specification of the number of PF keys controls the particular set of PF key definitions currently in use and affects the PF Key Definitions and Labels panel, which is displayed for the PF Keys option (0.3).

In the following cases, ISPF automatically senses the terminal type and the number of PF keys:

- If the screen size is larger than 24 lines, determined when you log on, ISPF sets the terminal type to 3278.
- If you press a PF key greater than 12, ISPF sets the terminal type to 3278 and sets the number of PF keys to 24.

ISPF automatically sets or changes the number of PF keys:

- To 12, if the terminal type is 3277.
- To whatever was stored from your last ISPF session, if the terminal type is 3278. For a new user, the number of keys is initialized to 12.
- To 24, if you press a PF key greater than 12.

ISPF cannot sense the number of PF keys if you switch from a 3278 with 24 PF keys to a 3278 with 12 PF keys. If you switch, you must inform ISPF of the number of PF keys by using the Terminal option (0.1) or the PF Keys option (0.3). Otherwise, the incorrect set of stored key definitions is used.

INPUT FIELD PAD

Specification of a pad character controls the first padding of panel input fields, including selection panels, but not the data portion of an Edit display. Within Edit mode, you can control null or blank padding with Edit commands. The pad character you specify can be a B to show a blank, N to show a null, or any non-alphanumeric character. The character you choose for input

field padding must not be the same as the character you choose for the command stacking delimiter.

COMMAND DELIMITER

You can stack commands on the COMMAND line of any panel by separating them with a delimiter. The default delimiter is the semicolon. If you want to change the delimiter, you can do so on this panel by specifying in the COMMAND DELIMITER field the character you want.

Stacking allows you to enter, for example:

```
COMMAND ==>      DEPT 16 DATA
```

which finds the characters DEPT and then switches to hexadecimal mode.

The character you choose for the command stacking delimiter must not be the same as the character you have chosen for input field padding.

The Browse and Edit FIND and CHANGE commands do not work with a search argument that contains the command delimiter, even if string delimiters are used.

SCREEN FORMAT

Specification of screen format applies to 3278 Model 5 and to 3290 terminals only; it is ignored for other types of terminals. If you specify DATA, ISPF automatically uses the larger “default” mode characters (24 lines by 80 characters) or the smaller “native” mode characters (27 lines by 132 characters), depending on the width of the data to be displayed. If you specify STD or MAX, the mode is not switched.

For a 3290 terminal, you can specify PART to use the maximum available screen size. If the 3290 is configured for partitions, PART tells ISPF to format the screen in one of four ways, depending on the 3290 configuration:

- 31 rows by 80 columns
- 62 rows by 80 columns
- 31 rows by 160 columns
- 62 rows by 160 columns.

Also, the SPLITV command is enabled when the 3290 is configured for hardware partitions. If the 3290 is not configured for hardware partitions, the default mode depends on the definition of the terminal in your system.

Note: The SPLITV function is not active if the data being displayed on a screen is more than 80 characters wide.

Log/List (Option 0.2)

When you select this option, the Log and List Defaults panel shown in Figure 19 is displayed to allow you to specify default processing for log and list data sets, lines per page, and allocation parameters.

```

----- LOG AND LIST DEFAULTS -----
COMMAND ===>

LOG DATA SET DEFAULT OPTIONS          LIST DATA SET DEFAULT OPTIONS
-----
Process option    ===> _               Process option    ===>
SYSOUT class     ===> A               SYSOUT class     ===> A
Local printer ID  ===>                Local printer ID  ===>
Lines per page    ===> 60             Lines per page    ===> 60
Primary pages     ===> 10             Primary pages     ===> 100
Secondary pages   ===> 10             Secondary pages   ===> 200

VALID PROCESS OPTIONS:
PD - Print data set and delete          D - Delete data set (without printing)
K - Keep data set (append subsequent information to same data set)
KN - Keep data set and allocate new data set

JOB STATEMENT INFORMATION:             (Required for system printer)
====> //HOSTETLA JOB (ACCOUNT),'NAME'
====> //*
====> //*
====> /*

```

Figure 19. Log and List Defaults Panel

The fields on the Log and List Defaults panel are:

Process option

Tells PDF how to process the list and/or log data sets. The valid processing options are:

- | | |
|-----------|---|
| PD | Print the data set and then delete it. |
| D | Delete the data set without printing it. |
| K | Keep the data set and allocate it in the next session. |
| KN | Keep the data set, but allocate a new data set in the next session. |

SYSOUT class

Destination of printed data set. Required only if data set is to be printed and Local printer ID is not specified.

Local printer ID

Destination of printed data set. Required only if data set is to be printed and SYSOUT class is not specified.

Lines per page

Normal values for lines per page are:

- | | |
|-----------|------------------------------------|
| 60 | For printing six lines per inch. |
| 80 | For printing eight lines per inch. |

Primary pages

The expected number of printout pages. This value is automatically converted by ISPF to the appropriate number of blocks before allocating space for the log and list data sets.

Secondary pages

The expected number of printout pages. This value is automatically converted by ISPF to the appropriate number of blocks before allocating space for the log and list data sets.

JOB STATEMENT INFORMATION

Required for background printing and language processing. See "Job Statement Information" on page 55 for more information.

Log and List Data Set Default Options

The ISPF log data set is created the first time you do an action that results in a log message, such as saving edited data or submitting a job to the batch machine. The ISPF list data set is created the first time you request a print function. The first values shown are:

LOG DATA SET DEFAULT OPTIONS		LIST DATA SET DEFAULT OPTIONS	
-----		-----	
SYSOUT class	==> A	SYSOUT class	==> A
Lines per page	==> 60	Lines per page	==> 60
Primary pages	==> 10	Primary pages	==> 100
Secondary pages	==> 10	Secondary pages	==> 200

ISPF provides no defaults for the other parameters on this panel. The entries you specify for the log and list data set default options are used when you end ISPF by using either the Exit option (X) or the RETURN command. The following rules apply:

- If you leave the process options fields blank or specify D, K, or KN, you must always specify lines per page for both data sets, and primary and secondary pages for the list data set.
- If you specify PD, you must also specify SYSOUT class and lines per page for both data sets, and primary and secondary pages for the list data set. You can specify different SYSOUT classes, but only one job is submitted for printing both data sets. For routing the data sets to a local printer, specify a printer ID and leave the SYSOUT class field blank.

If you do not enter job statement information, TSO prompts you when you leave PDF.

You cannot leave the Log and List Defaults panel until you comply with the preceding rules. Once you leave this panel, if a log or list data set is created and the Process option field for that data set is blank, the Specify Disposition of Log and List Data Sets panel is displayed if any of the following occur:

- The Exit option (X) is entered.
- The jump function with the Exit option (=X) is entered outside Dialog Test (option 7).
- The END or RETURN command is entered from the ISPF/PDF Primary Option Menu.

See “Ending PDF or a PDF Function” on page 22 for information about using the Specify Disposition of Log and List Data Sets panel.

After you review the parameters on this panel and change them as necessary, enter the END command to return to the previous panel.

Changing Primary and Secondary Page Values

If you change the primary/secondary allocation parameters after PDF allocates the data sets, the new values take effect the next time you enter ISPF.

For the log data set, you can specify a primary allocation of 0 (zero) to prevent allocation and generation of the log. You can avoid generation of the list data set by not requesting any print functions. If you change the defaults to prevent allocation of the log data set, and the log is already allocated, you must delete the log data set when you end ISPF to stop generating the log.

If you use the Dialog Test option (7), you should allocate the log data set. Dialog Test writes trace data to the log when you request it. Also, if Dialog Test finds an unexpected condition, problem data is written to the log.

PF Keys (Option 0.3)

The PF Key Definitions and Labels panels allow you to assign ISPF commands and labels to PF keys. When you enter the KEYS command or select the PF Keys option (0.3), one of two panels is displayed. The panel displayed depends on whether your terminal has 12 PF keys or 24 PF keys.

Terminals with 12 PF Keys

If your terminal has 12 PF keys, the panel shown in Figure 20 is displayed.

```
----- PF KEY DEFINITIONS AND LABELS -----
COMMAND ----> _
NUMBER OF PF KEYS ----> 12                      TERMINAL TYPE ----> 3278

PF1 ----> HELP
PF2 ----> SPLIT
PF3 ----> END
PF4 ----> RETURN
PF5 ----> RFIND
PF6 ----> RCHANGE
PF7 ----> UP
PF8 ----> DOWN
PF9 ----> SWAP
PF10 ----> LEFT
PF11 ----> RIGHT
PF12 ----> RETRIEVE

PF1 LABEL ---->          PF2 LABEL ---->          PF3 LABEL ---->
PF4 LABEL ---->          PF5 LABEL ---->          PF6 LABEL ---->
PF7 LABEL ---->          PF8 LABEL ---->          PF9 LABEL ---->
PF10 LABEL ---->        PF11 LABEL ---->          PF12 LABEL ---->

Press ENTER key to save changes.  Enter END command to save changes and exit.
```

Figure 20. PF Key Definitions and Labels Panel (12 PF Keys)

Terminals with 24 PF Keys

For terminals with 24 PF keys, the first panel displayed shows the “primary” keys, PF13-PF24. When you press the ENTER key, a panel showing the “alternate” keys, PF1-PF12, is displayed. You can alternate between the two panels by continuing to press the ENTER key. See Figure 21 on page 71.

Assigning ISPF/PDF Commands

The PF key command assignments shown in Figure 20 and in the first part of Figure 21 on page 71 are the default assignments distributed with PDF. See “ISPF Commands Assigned to PF Keys” on page 11 for definitions of these commands.

----- PF KEY DEFINITIONS AND LABELS - PRIMARY KEYS -----

COMMAND ----> _

NUMBER OF PF KEYS ----> 24

TERMINAL TYPE ----> 3278

PF13 ----> HELP

PF14 ----> SPLIT

PF15 ----> END

PF16 ----> RETURN

PF17 ----> RFIND

PF18 ----> RCHANGE

PF19 ----> UP

PF20 ----> DOWN

PF21 ----> SWAP

PF22 ----> LEFT

PF23 ----> RIGHT

PF24 ----> RETRIEVE

PF13 LABEL ----> PF14 LABEL ----> PF15 LABEL ---->

PF16 LABEL ----> PF17 LABEL ----> PF18 LABEL ---->

PF19 LABEL ----> PF20 LABEL ----> PF21 LABEL ---->

PF22 LABEL ----> PF23 LABEL ----> PF24 LABEL ---->

Press ENTER key to display alternate keys. Enter END command to exit.

----- PF KEY DEFINITIONS AND LABELS - ALTERNATE KEYS -----

COMMAND ----> _

NOTE: The definitions and labels below apply only to terminals with 24 PF keys.

PF1 ----> HELP

PF2 ----> SPLIT

PF3 ----> END

PF4 ----> TSO LISTALC ST

PF5 ----> RFIND

PF6 ----> RCHANGE

PF7 ----> UP

PF8 ----> DOWN

PF9 ----> SWAP

PF10 ----> LEFT

PF11 ----> RIGHT

PF12 ----> TSO SUBMIT NOTIFY

PF1 LABEL ----> PF2 LABEL ----> BLANK PF3 LABEL ---->

PF4 LABEL ----> DATASETS PF5 LABEL ----> FIND PF6 LABEL ----> CHANGE

PF7 LABEL ----> NOSHOW PF8 LABEL ----> NOSHOW PF9 LABEL ---->

PF10 LABEL ----> PF11 LABEL ----> PF12 LABEL ----> SUBMIT

Press ENTER key to display primary keys. Enter END command to exit.

Figure 21. PF Key Definitions and Labels Panels (24 PF Keys)

Before you change your PF key assignments, you should verify the terminal type and the number of PF keys, either 12 or 24. The terminal type must be one of the following:

- 3277, 3277A, or 3277KN
- 3278, 3278A, 3278T, 3278CF, or 3278KN
- 3290A.

For terminals with 24 PF keys, PF keys 1-12 have the same defaults as PF keys 13-24. Since the default settings are the commands you are likely to use most often, for ease of use we recommend the following:

- Use the primary PF keys for the default commands.
- Redefine the alternate PF keys to specific operations for special situations, such as when using Edit, Browse, or Dialog Test.

PF Keys (Option 0.3)

In this context, *alternate* means PF keys that require the use of the Alt key. For example, to use PF keys 1-12 on some terminals, you must press and hold the Alt key while pressing a PF key; PF keys 13-24 on these terminals are the *primary* PF keys because they are used *without* the Alt key.

You can assign the following types of commands:

- System commands, such as HELP or END
- Commands that are significant within a particular function or environment, such as the Edit FIND and CHANGE commands
- Line commands, such as Edit or Dialog Test I or D commands.

You can define or change a PF key function simply by equating the key to a command. For example:

```
PF9   ===>  
PF12  ===>
```

In the example, PF9 is equated to an Edit command and PF12 represents the system-defined PRINT command.

If you enter a blank for any PF key definition, the key is restored to its PDF default. The default PF key definitions are discussed under "ISPF Commands Assigned to PF Keys" on page 11.

To assign a line command to a PF key, you must begin the PF key definition with a colon. When you press the PF key, the colon is removed and the command to which the key is equated is inserted in the first input field in the line at which the cursor is currently positioned. For example:

```
PF10  ===> :  
PF11  ===> :
```

In this example, PF keys 10 and 11 have been equated to the Edit TS (Text Split) and TF (Text Flow) line commands, respectively.

Assigning Labels

The LABEL fields shown in Figure 20 on page 70 and Figure 21 on page 71 apply only when the PFSHOW command has been issued with the ON (default) or TAILOR option. Instead of using the actual PF key definitions, the displayed representations of PF key definitions can be labels that you define.

If labels are not assigned, the definition displayed for each PF key consists of the first eight characters of the PF key definition. However, the LABEL fields let you display significant words of eight characters or less, instead of the first eight characters of a lengthy PF key definition, which may be meaningless when partially displayed.

If the value assigned to a label is BLANK, the PFSHOW command displays the PF key number and an equal sign (=), but the PF key definition does not appear. Why, then, use the BLANK label? Well, a dialog developer, for

example, might use this label if a PF key is not defined or is meaningless to the user, but the developer still wants the PF key number to appear sequentially in the PF key definition lines.

No PF key information, not even the number, appears if the label value for that key is NOSHOW.

The second part of Figure 21 on page 71 shows how the PF Key Definitions and Labels panel can be used to assign definitions and labels. In this example, PF4 and PF12 have been equated to TSO commands and their respective labels have been assigned significant abbreviated versions of these commands. Labels for several other PF keys are defined in the figure as well.

Figure 22 shows the effect of the PFSHOW command for the PF key definitions and labels specified in the example.

Note: To duplicate this example, you must use the PFSHOW command with the TAILOR option and specify that alternate PF keys are to be displayed. The reason for this is that the labels were defined on the PF Key Definitions and Labels - Alternate Keys panel.

For more information, see the description of the PFSHOW command in *Dialog Management Guide* or in the online tutorial.

```

----- DIALOG TEST PRIMARY OPTION MENU -----
OPTION ==> _

1- FUNCTIONS      - Invoke dialog functions/selection menus
2- PANELS         - Display panels
3- VARIABLES      - Display/set variable information
4- TABLES        - Display/modify table information
5- LOG            - Browse ISPF log
6- DIALOG SERVICES - Invoke dialog services
7- TRACES         - Specify trace definitions
8- BREAKPOINTS    - Specify breakpoint definitions
T- TUTORIAL        - Display information about Dialog Test
X- EXIT           - Terminate dialog testing

Enter END command to terminate dialog testing.

PF 1=HELP      2-      3=END      4=DATASETS  5=FINO      6=CHANGE
PF 9=SWAP      10=LEFT  11=RIGHT  12=SUBMIT

```

Figure 22. Sample Screen with PF Key Definition Lines

Display (Option 0.4)

When you select this option, the Display Characteristics panel shown in Figure 23 is displayed.

```
----- DISPLAY CHARACTERISTICS -----
COMMAND ===> _
COMMAND LINE PLACEMENT ===> ASIS  (ASIS - Display as shown
                                     in panel definition.)
                                     (BOTTOM - Display as the last
                                     line on the screen or
                                     as the last line above
                                     the split line.)
```

Figure 23. Display Characteristics Panel

The only field on this panel is:

COMMAND LINE PLACEMENT

Specifies where the COMMAND and long message lines are to be displayed on the screen. You have two choices:

ASIS If you specify ASIS, the screen is displayed as specified in the panel definitions for the various panels displayed during the session. Since the COMMAND lines on some of your panels cannot be moved to the bottom, you may want to keep the COMMAND line placement for existing dialogs set to ASIS.

BOTTOM If you specify BOTTOM:

- The COMMAND line is moved to the last line of the logical screen.
- Each line that followed the original COMMAND line is shifted *up* one line.
- The long message overlays the line above the new COMMAND line location of the logical screen.

On table display panels, the line above the new COMMAND line is always overlaid by a blank line and used as the long message line. This prevents

table display data from being overlaid by long message data. For additional information about restrictions on the use of the BOTTOM placement, see *Dialog Management Guide*.

List (Option 0.5)

The List option (0.5) allows you to specify the characteristics of the records to be contained in the list data set when it is allocated.

When you specify the List option (0.5), the List Data Set Characteristics panel shown in Figure 24 is displayed.

----- LIST DATA SET CHARACTERISTICS -----

COMMAND ----> _

RECORD FORMAT ----> FBA (FBA VBA)

LOGICAL RECORD LENGTH ----> 121 (Any record length for fixed)
(Greater than 4 for variable)

LINE LENGTH ----> 120 (80 - 160)

Figure 24. List Data Set Characteristics Panel

When you reset the characteristics, their new values take effect *immediately* unless the list data set has already been allocated. In that case, the new values are used in the next ISPF session. These values are saved in your user profile, which ISPF automatically builds and maintains across sessions.

These are the fields on the List Data Set Characteristics panel:

RECORD FORMAT

Specifies the format and characteristics of the records in the list data set. The allowable record formats are:

- FBA

Fixed-length records that contain ANSI-defined printer control characters.
- VBA

Variable-length records that contain ANSI-defined printer control characters.

The default setting for the list data set's record format is FBA.

LOGICAL RECORD LENGTH

Specifies the length, in bytes, of fixed-length records, or the maximum length allowed for variable-length records. The default value for the list data set's logical record length is 121. This value represents one ANSI-defined control character and 120 bytes of data to be printed.

LINE LENGTH

Specifies the length of the logical line to be printed. If you specify a line length greater than the list data set's logical record length, the data is truncated. The range of allowable lengths is from 80 bytes to 160 bytes. The default value for the list data set's line length is 120.

Printing Panels with Long Lines

You can use the List option (0.5) to print panels whose line lengths would not otherwise be supported by the available printing facilities. For example:

- A panel of 160 bytes in width is to be printed, but printing capabilities allow only 132 bytes.

You should specify the following:

RECORD FORMAT ===> or

LOGICAL RECORD LENGTH ===>

Note: A logical record length of 132 allows for two ANSI-defined control characters.

LINE LENGTH ===>

The first page of output would contain the first 130 bytes of the panel and the second page would contain the last 30 bytes.

- A panel 132 bytes in width is to be printed; the line length is supported by printing facilities.

You should specify the following:

RECORD FORMAT ===> or

LOGICAL RECORD LENGTH ===>

Note: A logical record length of 133 allows for one ANSI-defined control character.

LINE LENGTH ===>

The complete panel would be printed out on one page of output.

- A panel of 80 bytes in width is to be printed; default values for the LIST parameters are to be used.

The complete panel would be printed out on one page of output.

Graphic (Option 0.6)

When you select option 6 from the ISPF Parameter Options panel, the Print Graphics Parameters panel, shown in Figure 25, allows you to tell GDDM:

- The name of the printer to which graphic display output is to be routed
- How the graphics aspect ratio (relationship to displayed screen image) is to appear on the printed output.

The parameters entered on this panel are used whenever you enter the PRINTG command. Unlike the other ISPF print commands, such as PRINT and PRINTL, the PRINTG command allows you to print a panel *and* its graphic area.

----- PRINT GRAPHICS PARAMETERS -----

COMMAND ==> _

FAMILY PRINTER TYPE ==> 2 (2 - QUEUED PRINTER)

DEVICE NAME ==> * (FOR FAMILY PRINTER TYPE 2)

ASPECT RATIO ==> 0 (0 - PRESERVE GRAPHIC ASPECT
 1 - PRESERVE POSITIONAL)

Figure 25. Print Graphics Parameters Panel

The Print Graphics Parameters panel uses the following fields:

FAMILY PRINTER TYPE

The printer type of the destination device. The only valid value is 2, which represents a queued printer.

DEVICE NAME

The destination of the printed output. This is the VTAM node name for the printer and is defined by your installation.

ASPECT RATIO

Enter one of the following values to tell the printer to:

- 0
- Preserve the graphics aspect ratio for a “true” picture. The ratio of the graphic area width to its height is the same on the printed document as in the displayed view. Figure 26 on page 79 and Figure 27 on page 79 show the difference between how a graphic may appear when displayed on the screen and when printed using aspect

ratio parameter 0. This value is used as the default if you leave this field blank.

- 1 Preserve the positional relationship between the graphic and the alphanumeric characters outside the graphics area. The printed graphic is either stretched or compressed horizontally to be aligned with characters outside the graphics area. Figure 26 and Figure 28 on page 80 show the difference between how a graphic may appear when displayed on the screen and when printed using aspect ratio parameter 1.

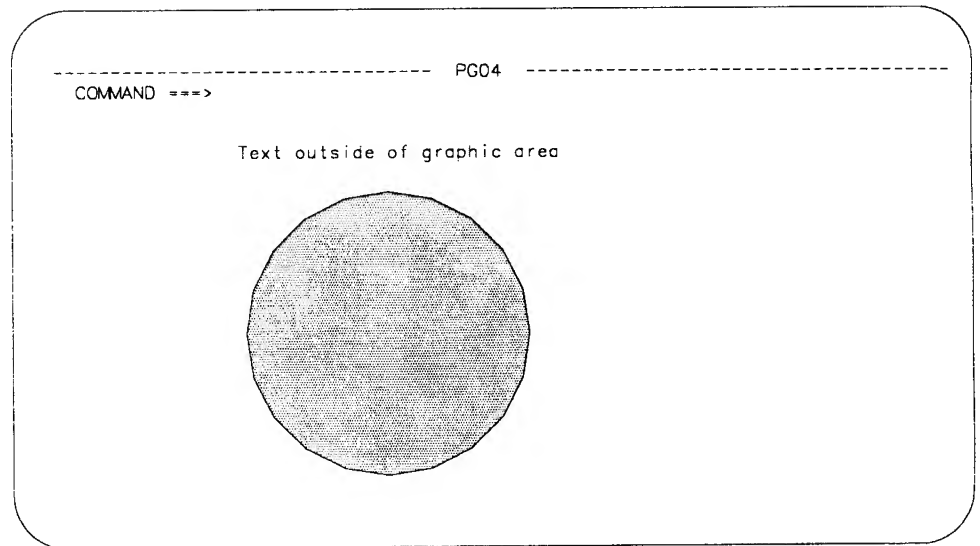


Figure 26. Screen Containing Graphics to be Printed

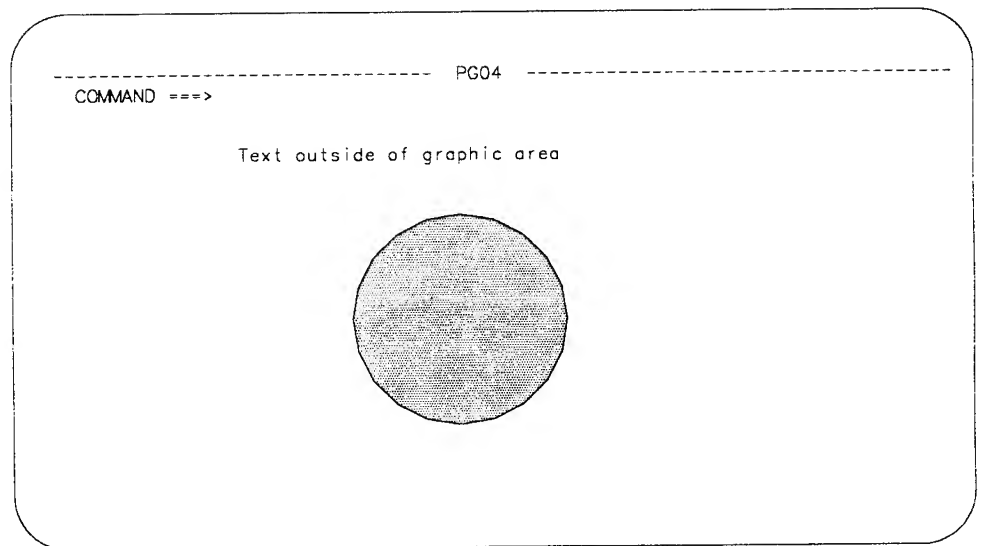


Figure 27. Example of Using Aspect Ratio Parameter 0

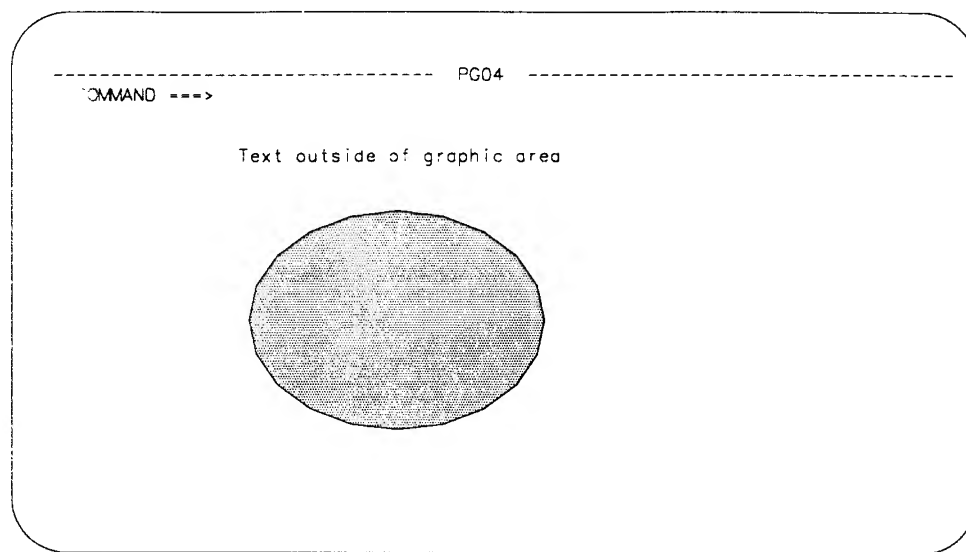


Figure 28. Example of Using Aspect Ratio Parameter 1

Environ (Option 0.7)

The Environ option (0.7) lets you set default values for the ISPF ENVIRON command parameters. ISPF provides the ENVIRON command to help you in gathering data that can be helpful in diagnosing problems, thus reducing service time. The ISPF session does not have to be running in any ISPF TEST/TRACE mode when you use the ENVIRON command. You can issue the ENVIRON command at any time during an ISPF session.

The ENVIRON command can help you:

- Produce system ABEND dumps when not running in ISPF TEST mode (ENBLDUMP parameter)
- Trace the TPUT, TGET, and PUTLINE buffers and get dump information for TPUT and TGET errors (TERMTRAC parameter)
- Gather terminal status information (TERMSTAT parameter).

See *Dialog Management Guide* for complete information about the ENVIRON command.

When you select option 0.7 or enter the ENVIRON command without any parameters, the panel shown in Figure 29 is displayed. This panel includes the current values of the ENVIRON command parameters (ENBLDUMP and TERMTRAC) and the ddname, if any, allocated for a dump data set. The values can be changed by entering new values directly on the panel.

```

----- ISPF ENVIRON COMMAND SETTINGS -----
COMMAND ----> _

ENBLDUMP Setting ----> OFF      (ON   - Enables a dump for a subtask abend,
                                - when not in ISPF TEST mode. SYSUDUMP,
                                - SYSMDUMP, or SYSABEND must be
                                - allocated to obtain the dump.
                                OFF  - Disables ON setting.)

TERMTRAC Setting ----> OFF      (ON   - Enables terminal tracing.
                                ERROR - Enables terminal tracing, issues SNAP
                                - when a terminal error is encountered.
                                OFF  - Disables ON or ERROR setting.
                                DUMP - May only be specified as an ENVIRON
                                - command parameter.)

      DDNAME ----> ISPSNAP      (DDNAME for TERMTRAC ON, ERROR, or DUMP.)

Invoke TERMSTAT ----> NO       (YES  - Type YES and press the ENTER key to
                                - run ENVIRON TERMSTAT.
                                QUERY - Type QUERY and press the ENTER key to
                                - run ENVIRON TERMSTAT QUERY.
                                NO   - Do not run ENVIRON TERMSTAT.)

```

Figure 29. ISPF ENVIRON Command Settings Panel

The fields on the ISPF ENVIRON Command Settings panel are:

ENBLDUMP Setting

Specifying the ENBLDUMP parameter allows ISPF to produce an ABEND dump if a subtask abnormally ends when ISPF is not running in TEST mode (as required before ISPF Version 2.3).

The following parameters are valid for the ENBLDUMP Setting field:

ON Specifies to ISPF that a dump for the ABENDING subtask is to be generated and control returned to TSO. ISPF execution is not resumed. This is the default.

When running in ISPF TEST mode, issuing ENVIRON ENBLDUMP has no effect on dump processing.

OFF Cancels the effect of the ON status.

TERMTRAC Setting

Specifying the TERMTRAC parameter allows you to trace all terminal input and output data (TPUT, TGET, PUTLINE) during an ISPF session.

Before using the TERMTRAC option, you must define to ISPF the ddname for the data set to be used for the System Network Analysis Program (SNAP) macro, which ISPF invokes to provide data stream dumps. This ddname should have a disposition of MOD to assure that no trace data is lost. It can be defined by specifying it on the panel. See DDNAME on page 83 for more information.

You cannot enter the DUMP parameter in the TERMTRAC Setting field. See "Using the DUMP Parameter" on page 84 for information about using this parameter.

The following parameters are valid for the TERMTRAC Setting field:

ON Starts TPUT, TGET, and PUTLINE buffer tracing of the terminal data stream. All data is retained in a 24K buffer provided by ISPF. Whenever the trace buffer is full, ISPF issues a SNAP macro to dump the trace buffer only. For more information about the SNAP macro, see *MVS OS/VS2 Supervisor Services and Macro Instructions* or *MVS/XA Supervisor Services and Macro Instructions*.

ERROR Begins tracing of the TPUT, TGET, and PUTLINE buffers. Also, it causes ISPF to begin an MVS SNAP dump if a TPUT or TGET error occurs.

ISPF issues the SNAP macro on the first occurrence of a TPUT failure. ISPF makes three consecutive attempts to correct a TPUT error.

OFF Resets active ENVIRON TERMTRAC ON and ENVIRON TERMTRAC ERROR commands. If ENVIRON TERMTRAC is active, ISPF issues a SNAP macro.

DDNAME The name of the data set that is to receive the dump(s) from the SNAP macro. If you leave this field blank, ISPF uses ISPSNAP as the default. Therefore, to use a name other than ISPSNAP, you must fill in this field before invoking the ENVIRON command with either TERMTRAC ON, TERMTRAC ERROR, or TERMTRAC DUMP (see "Using the DUMP Parameter" on page 84).

You must follow the data set characteristics guidelines defined by MVS for the SNAP macro. For DCB information that can be specified for the SNAP ddname, see *MVS OS/VS2 Supervisor Services and Macro Instructions* or *MVS/XA Supervisor Services and Macro Instructions*.

Invoke TERMSTAT

Specifying the TERMSTAT option of the ENVIRON command allows you to collect information about the characteristics of the terminal you are using and the line to which it is attached. The information is divided into three parts:

- A list of terminal characteristics as defined in ISPF variables. This list defines what ISPF thinks your terminal characteristics are.
- A list of terminal characteristics as defined within TSO.
- A list of structured fields that apply only to terminals with extended data stream (EDS) capability.

The following parameters are valid for this field:

YES Invokes the terminal status dialog, which sends the settings of various terminal I/O variables to the terminal and to the ISPF log data set.

QUERY Forces terminals that support EDS, such as an IBM 3279, but are not connected to an EDS port to include a list of structured fields in the information returned.

If your terminal does *not* support EDS, such as an IBM 3277, the QUERY parameter causes you to receive an ORDER CHECK STREAM error.

NO Disables the terminal status dialog.

Using the DUMP Parameter

You cannot enter the DUMP parameter in the TERMTRAC Setting field. This parameter is not to be used except when entered with the ENVIRON command on the COMMAND line, such as:

The DUMP parameter causes ISPF to immediately issue a SNAP macro, but *only if ENVIRON TERMTRAC ON or ENVIRON TERMTRAC ERROR is active*. The resulting dump includes the storage trace buffer, the current task control block (TCB), all system control program information, and all problem program information. The MVS SNAP macro definition provides more specific information about the areas dumped when all system control program and problem program information is requested.

Note that:

- This command execution does *not* turn off terminal data stream tracing if it is active at the time.
- The next entry is written to the top of the terminal data buffer if the SNAP was successful; otherwise, tracing continues immediately after the last trace buffer entry.

CHAPTER 4. BROWSE (Option 1)

This chapter describes Browse, option 1 on the ISPF/PDF Primary Option Menu. This option allows you to display, but not change, data stored in ISPF libraries or TSO data sets. The chapter includes descriptions of the commands you can use while browsing a data set.

Chapter 4. Browse (Option 1)

The Browse option (1) allows you to display source data and listings stored in ISPF libraries or other partitioned or sequential data sets that have the following characteristics:

- Record Format (RECFM):
 - Fixed, variable (non-spanned), or undefined
 - Blocked or unblocked
 - With or without printer control characters.
- Logical Record Length (LRECL):
 - For fixed-length records, up to 32,760 characters
 - For variable-length records, up to 32,756 characters.

Browsing a Data Set

When you select the Browse option, the Browse - Entry Panel shown in Figure 30 is displayed.

```

----- BROWSE - ENTRY PANEL -----
COMMAND ---->

ISPF LIBRARY:
PROJECT ----> ISPFDEMO
GROUP ----> MYLIB
TYPE ----> PL1
MEMBER ----> _ (Blank or pattern for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ---->
VOLUME SERIAL ----> (If not cataloged)

DATA SET PASSWORD ----> (If password protected)

MIXED MODE ----> NO (Specify YES or NO)

FORMAT NAME ---->
  
```

Figure 30. Browse - Entry Panel

All the fields on this panel are explained in Chapter 2, “Libraries and Data Sets.” For information about browsing libraries that are controlled under Library Management Facilities, see *Library Management*.

Browse Data Display

Some of the fields on this panel may retain information entered previously, such as PROJECT, GROUP, and TYPE values. When you fill in the fields on this panel and press the ENTER key, you see either a member selection list or a Browse data display. See "Using Member Selection Lists" on page 37 if you need information about displaying a member list.

Browse Data Display

A Browse data display is shown in Figure 31.

```
BROWSE -- ISPFDEMO.XXX.COBOL(CBLMAIN) - 01.01 ----- LINE 00000000 COL 001 080
COMMAND ==> _ SCROLL ==> PAGE
***** TOP OF DATA *****
IDENTIFICATION DIVISION.
PROGRAM-ID. 'F20D1000'.
DATE-COMPILED. OCT. 20, 1985
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. IBM-370.
OBJECT-COMPUTER. IBM-370.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT OLD-COMREC ASSIGN TO DA-S-DD1.
    SELECT D1-REPORTS ASSIGN TO UR-S-D1OUT.
    SELECT OPTION-CARD-FILE ASSIGN TO UR-S-SYSIN.
DATA DIVISION.
FILE SECTION.
FD OLD-COMREC
   LABEL RECORD IS STANDARD
   RECORDING MODE IS F
   BLOCK CONTAINS 0 RECORDS
   DATA RECORD IS COMREC1.
01 COMREC1.
02 DUMMY PICTURE X(520).
```

Figure 31. Browse - Data Display

Each character in the data that cannot be displayed is changed on the display to either a period or a character that you have specified. Using the DISPLAY command, you can specify whether printer carriage control characters are to be treated as part of the data, and thus displayed.

During Browse, four-way scrolling is available via the scroll commands. You can also use the FIND and LOCATE commands to scroll to a particular character string, line number, or symbolic label.

Whenever you enter a command, such as FIND or one of the scroll commands, that puts the cursor under a character string in the data set, PDF highlights that character string. This highlighting occurs whether you type the command on the COMMAND line and press the ENTER key or press a PF key to which the command is assigned.

Ending Browse

To end a Browse data display, enter the END command. This returns you to the previous panel, which is either a member list display or the Browse - Entry Panel. If a member list is displayed, the name of the member you just browsed is at the top of the list. You can select another member from the list or enter the END command again to return to the Browse - Entry Panel.

When the Browse - Entry Panel is displayed again, you can select another data set or member, or you can enter the END command to return to the ISPF/PDF Primary Option Menu.

Browse Primary Commands

You can prefix any Browse command with an ampersand to keep the command displayed on the COMMAND line after the command has executed. This technique allows you to repeat similar commands without re-entering the data. For example, if you enter:

```
COMMAND ===>
```

the command is displayed after the string has been found, which allows you to then change the parameter and issue another FIND command.

Browse provides seven functions, described in the following sections, each of which is controlled by a command that you can type in the COMMAND field:

BROWSE	Browsing recursively.
COLUMNS	Identifying columns.
DISPLAY	Controlling the display.
FIND	Finding character strings.
HEX	Displaying data in hexadecimal format.
LOCATE	Locating lines.
RESET	Removing the column-identification line.

BROWSE - Browsing Recursively

The BROWSE command allows you to browse another member of the same data set. It also allows you to browse any other data set without ending your current Browse session.

Browse Primary Commands

The BROWSE command has the following format:

```
BROWSE [member]
```

where:

member An optional member of the ISPF library or “other” partitioned data set that you are currently browsing.

For example, if you were browsing a member of library ISPFDEMO.XXX.COBOL, you could enter the following command to display the panel shown in Figure 31 on page 86:

If you do not specify a member name, the Browse Command Panel is displayed. This panel, except for its title, is the same in both appearance and function as the Browse - Entry Panel shown in Figure 30 on page 85.

You end a nested Browse session the same as you would a normal one. When you end the nested Browse session, the current Browse session resumes.

COLUMNS - Identifying Columns

You can use the COLUMNS command to provide a temporary indication of where columns occur on the panel. This command displays a column-identification line on the first line of the data area. The command has the following format:

```
COLUMNS on | off
```

where:

on Displays the column-identification line. This is the default.
off Removes the column-identification line from the display.

Note: You can also remove the column-identification line by entering the RESET command.

An example of the column-identification line is shown in Figure 32 on page 89. The digits on the identification line show the “tens” positions: “1” shows column 10, “2” shows column 20, and so forth. The plus signs (+) show the “fives” positions.


```

BROWSE -- ISPFDEMO.XXX.COBOL(CBL) - 01.01 ----- LINE 00000022 COL 001 080
COMMAND ==> -                                     SCROLL ==> PAGE
-----1-----2-----3-----4-----5-----6-----7-----8
002200 FD  D1-REPORTS
002300     LABEL RECORD IS OMITTED
002400     RECORDING MODE IS F
002500     BLOCK CONTAINS 133 CHARACTERS
002600     DATA RECORD IS D1-LINE.
002700 01  D1-LINE.
002800     02 COLUMNS          PICTURE X(133).
002900 FD  OPTION-CARD-FILE
003000     LABEL RECORD IS OMITTED
003100     BLOCK CONTAINS 1 RECORDS
003200     RECORDING MODE IS F
003300     DATA RECORD IS O-C.
003400 01  O-C.
003500     02 DUMMY            PICTURE X(80).
003600 WORKING-STORAGE SECTION.
003700     77 OP-SUB             PICTURE S99 COMPUTATIONAL VALUE 0.
003800     77 PREV-DEVICE-TYPE-CODE PICTURE X VALUE 'I'.
003900     77 PREV-ACTV-CODE       PICTURE 9 VALUE 0.
004000     77 PREV-PROB-CODE     PICTURE 9 VALUE 0.
004100     77 C-SWITCH           PICTURE X VALUE '0'.
004200     77 PREV-SYSTEM-CODE    PICTURE X VALUE SPACE.

```

Figure 32. Browse - Column-Identification Line

DISPLAY - Controlling the Display

The DISPLAY command allows you to display carriage control characters and characters that cannot normally be displayed. It has the following format:

```

DISPLAY {char}
        {NOCC | CC}

```

You must enter at least one parameter, but you can enter them in any order. If you enter only one parameter, the other parameter retains its current value.

- | | |
|-------------|--|
| char | The character you want to use to represent characters that cannot be displayed on the screen. It can be a single character, or a single character enclosed in apostrophes (') or quotation marks ("). If you specify a blank as the character, you <i>must</i> enclose it in apostrophes or quotation marks. |
| CC | Shows that carriage control characters are to be displayed and are to be considered part of the data. |
| NOCC | Shows that carriage control characters are not to be displayed and are not to be considered part of the data. |

Both operands are stored in your user profile and are in effect whenever you are using Browse. You need to re-enter the DISPLAY command only if you want to change one of the operands.

The first settings for display mode are period (.) and NOCC, but the carriage control character status has no effect if the data that you are browsing has no carriage control characters.

Browse Primary Commands

The following examples show three different ways to enter the DISPLAY command:

- To use blanks to represent characters that cannot be displayed, enter:
- To use “|” to represent characters that cannot be displayed, enter:
- To suppress the display of carriage control characters, enter:

FIND - Finding Character Strings

The FIND command lets you find a specified character string. The format of the FIND command is:

```
FIND string [NEXT | ALL | FIRST | LAST | PREV]
           [CHARS | PREFIX | SUFFIX | WORD]
           [col-1 [col-2]]
```

Note: FIND as a Browse command, shown here, has the same format as FIND as an Edit command, except the optional X/NX/EX and line range parameters are not included.

where:

string The character string you want to find.

NEXT | ALL | FIRST | LAST | PREV

Operands that define the starting point, direction, and extent of the search.

CHARS | PREFIX | SUFFIX | WORD

Operands that set the conditions for a character string match.

col-1 and col-2

Numbers that identify the columns the FIND command is to search.

You can separate the operands with blanks or commas and you can type them in any order, but col-2, if typed, must follow col-1. The string operand is required. The others are optional.

Specifying Find Strings

The string operand specifies the characters to be found. For examples of different string formats, see the description of the FIND command in *Edit and Edit Macros*.

The default is not to differentiate between uppercase and lowercase characters when *searching*. Except for the character (C) string, differences

between uppercase and lowercase strings are ignored. For example, this command:

successfully finds any of the following:

```
CONDITION NO. 1
Condition No. 1
condition no. 1
coNDitION nO. 1
```

Omitting String Delimiters:

Generally, you enter the strings without delimiters. For example, to find all occurrences of ABC, enter:

Using String Delimiters:

You must use delimiters if a string contains imbedded blanks or commas, or if a string is the same as a command keyword. You delimit strings with either apostrophes (') or quotation marks ("). For example, to find the next occurrence of "every one," enter:

The FIND command does not find the apostrophe or quotation mark string delimiters.

Starting Point, Direction, and Extent of Search

You can control the starting point, direction, and extent of the search by using one of the following operands:

- | | |
|--------------|---|
| NEXT | The scan starts at the first position after the current cursor location and searches ahead to find the next occurrence of the string. NEXT is the default. |
| ALL | The scan starts at the top of the data and searches ahead to find all occurrences of the string. A message in the upper-right corner of the screen shows the number of occurrences found. The second-level message that is displayed when you enter the HELP command shows which columns were searched. |
| FIRST | The scan starts at the top of the data and searches ahead to find the first occurrence of the string. |
| LAST | The scan starts at the bottom of the data and searches backward to find the last occurrence of the string. |
| PREV | The scan starts at the first position before the current cursor location and searches backward to find the previous occurrence of the string. |

Browse Primary Commands

If you specify FIRST, ALL, or NEXT, the direction of the search is forward; pressing the R_FIND PF key (PF5/17) finds the next occurrence of the designated string. If you specify LAST or PREV, the direction of the search is backward; pressing the R_FIND PF key finds the previous occurrence of the string. The other optional parameters remain in effect, as specified in the last FIND command. These parameters include CHARS, WORD, PREFIX, SUFFIX, and col-1, col-2.

The search proceeds until one or all occurrences of the string are found, or until the end of data is encountered. If the string is not found, one of the following actions takes place:

- If the FIND command was entered in the COMMAND field, a **NO string FOUND** message is displayed in the upper-right corner of the screen.
- If the FIND command was repeated using the R_FIND command, either a **BOTTOM OF DATA REACHED** message or a **TOP OF DATA REACHED** message is displayed, depending on the direction of the search. When these messages appear, you can press the R_FIND PF key again to continue the search by wrapping to the top or bottom of the data. If the string is still not found anywhere in the data, a **NO string FOUND** message is displayed.

Conditions for Character String Matches

The operands CHARS, PREFIX, SUFFIX, and WORD control the conditions for a successful “match” with the string based on whether the data string begins and/or ends with a non-alphanumeric character; that is, a special character or a blank. You can abbreviate PREFIX, SUFFIX, and CHARS to PRE, SUF, and CHAR, respectively.

In the following illustration, the underscored strings would be found and the non-underscored strings would be ignored:

```
CHARS 'DO' - DO DONT ADO ADOPT 'DO' +ADO (DONT) ADO-
PREFIX 'DO' - DO DONT ADO ADOPT 'DO' +ADO (DONT) ADO-
SUFFIX 'DO' - DO DONT ADO ADOPT 'DO' +ADO (DONT) ADO-
WORD 'DO' - DO DONT ADO ADOPT 'DO' +ADO (DONT) ADO-
```

If you do not specify an operand, the default is CHARS.

Use of Text Strings

Text strings are processed exactly the same as delimited strings. They are provided for compatibility with prior versions of the product.

Use of Character Strings

A character string, which may be used as a string parameter in a FIND command, requires that the search be satisfied by an exact character-by-character match. Lowercase alphabetic characters match only with lowercase alphabetic characters and uppercase alphabetic characters match only with uppercase.

For example, FIND C'XYZ' would find the characters XYZ only if they were in uppercase. FIND C'xyz' would find the characters xyz only if they were in lowercase.

If you specify a text string that contains any SO or SI characters, the string is considered a character string.

Use of Picture Strings

A picture string used as a string parameter in a FIND command allows you to search for a particular type of character, without regard for the specific character involved. You can use special characters within the picture string to represent the type of character to be found, as follows:

String Meaning

P'='	Any character.
P'␣'	Any nonblank character.
P'.'	Any nondisplayable (invalid) character.
P'0'	Any numeric character (0-9).
P'-'	Any non-numeric character.
P'@'	Any alphabetic character (upper- or lowercase).
P'<'	Any lowercase alphabetic character.
P'>'	Any uppercase alphabetic character.
P'\$'	Any special character (not alphabetic or numeric).

If an APL or TEXT keyboard is being used, the following additional character can be used in a picture string:

P'␣'	Any APL-specific or TEXT-specific character.
------	--

Only the special characters listed above are valid within a picture string, but the string can include alphabetic or numeric characters that represent themselves.

A DBCS subfield cannot be specified as the subject of a picture string for the FIND command.

Examples of picture strings:

P'###'	A string of three numeric characters.
P'␣ ␣'	Any two nonblank characters separated by a blank.
P'.'	Any nondisplayable character.
P' #'	A blank followed by a numeric character.
P'#AB'	A numeric character followed by 'AB'.

Browse Primary Commands

Examples of FIND commands using picture strings:

FIND P'.' Find next nondisplayable character.

FIND P'¬' 72 Find next nonblank character in column 72.

F P' ¬' 1 Find the next line with a blank in column 1 followed by a nonblank.

When you use the special characters '=' or '.' and a nondisplayable character is found, a hexadecimal representation is used in the confirmation message that appears in the upper-right corner of the screen. For example:

could result in the message **CHARS X'0205' FOUND**.

Column Limitations

The col-1 and col-2 operands allow you to search only a portion of each line, rather than the complete line. These operands, which are integers separated by a comma or by at least one blank, show the starting and ending columns for the search. The following rules apply:

- If you specify neither col-1 nor col-2, the search continues across all columns within the current boundary columns.
- If you specify col-1 without col-2, the string is found only if it starts in the specified column.
- If you specify both col-1 and col-2, the complete string, not just part of it, must be within the specified columns.

Using RFIND

The RFIND command, which is usually assigned to the PF5/17 key, lets you repeat the previous FIND command without retyping it. Therefore, you can use this command to find successive occurrences of the string specified in the last FIND command. You can also use the RFIND command to return to the top of your data and continue searching when the **BOTTOM OF DATA REACHED** message appears. If you type RFIND in the COMMAND field instead of using a PF key, you must position the cursor to the desired starting location *before* pressing the ENTER key.

If you specify a one-byte hexadecimal string as the FIND string and the string is found at the second byte of a double-byte character set (DBCS) character, hardware sets the cursor to the first byte of the character. If you then request RFIND, the same data is found again. To find the next occurrence of the string, you must move the cursor to the next character position before requesting RFIND again.

HEX - Displaying Data in Hexadecimal Format

The display of data in hexadecimal notation is controlled by the HEX command. The format of the command is:

```
HEX  [ON | OFF]
      [VERT | DATA]
```

You can specify the operands in any order.

ON Turns hexadecimal mode on.

OFF Turns hexadecimal mode off. If you omit this operand, **ON** is assumed.

VERT Valid only when hexadecimal mode is **ON**. Figure 33 on page 96 shows how **VERT** causes the hexadecimal representation to be displayed vertically, two rows per byte, under each character.

DATA Valid only when hexadecimal mode is **ON**. Figure 34 on page 96 shows how **DATA** causes the hexadecimal representations to be displayed as a string of hexadecimal characters, two per byte. Since the hexadecimal string is twice the length of the data string, it consumes two rows. If you omit this operand, **VERT** is assumed.

For example, the following command would display the hexadecimal notation vertically:

Three lines are displayed for each source line. The first line shows the data in standard character form. Figure 33 on page 96 shows the next two lines with the same data in vertical hexadecimal representation. A separator line is displayed between the two representations to make it easier for you to read the data.

To display the hexadecimal notation horizontally, use this command:

Figure 34 on page 96 shows the next two lines with the same data in data hexadecimal representation.

You can use the **FIND** command to find invalid characters or any specific hexadecimal character regardless of the setting of hexadecimal mode. See the syntax for picture strings and hexadecimal strings under the description of the **FIND** command in *Edit and Edit Macros*.

Browse Primary Commands

```
BROWSE -- FBROWN.PRIVATE.SCRIPT(BHEX) - 01.00 ----- LINE 00000000 COL 001 076
COMMAND ==> _ SCROLL ==> PAGE
***** TOP OF DATA *****

-----
:H3 ID=BRHEX SUBJECT='BROWSE COMMANDS - HEX'.
7CF4CC7CDCCE4EECDCE77CDDEEC4CDDDCDCE464CCE74
A83094E2985702421533ED296625036441542000857DB
-----
HEX - DISPLAYING DATA IN HEXADECEIMAL FORMAT
CCE464CCEDDCECDC4CCEC4CD4CCECCCCDCD4CDDDCDCE
8570004927318957041310950857145394130669413
-----
:P.THE DISPLAY OF DATA IN HEXADECEIMAL NOTATION IS CONTROLLED BY THE HEX
7D4ECC4CCEDDCE4DC4CCEC4CD4CCECCCCDCD4DDECECD4CE4CDDDDDDCC4CE4ECC4CCE
A7B38504927318066041310950857145394130563139650920365396335402803850857
-----
COMMAND.
CDDDCDC44
3644154B0
-----
THREE LINES ARE DISPLAYED FOR EACH SOURCE LINE.
```

Figure 33. Browse Hexadecimal Display - Vertical

```
BROWSE -- FBROWN.PRIVATE.SCRIPT(BHEX) - 01.00 ----- LINE 00000000 COL 001 076
COMMAND ==> _ SCROLL ==> PAGE
***** TOP OF DATA *****

-----
:H3 ID=BRHEX SUBJECT='BROWSE COMMANDS - HEX'.
7AC8F340C9C47EC2D9C8C5E740E2E4C2D1C5C3E37E7DC2D9D6E6E2C540C3D6D4D4C1D5C4E2406040
C8C5E77D4B
-----
HEX - DISPLAYING DATA IN HEXADECEIMAL FORMAT
C8C5E7406040C4C9E2D7D3C1E8C9D5C740C4C1E3C140C9D540C8C5E7C1C4C5C3C9D4C1D340C6D6D9
D4C1E3
-----
:P.THE DISPLAY OF DATA IN HEXADECEIMAL NOTATION IS CONTROLLED BY THE HEX
7AD74BE3C8C540C4C9E2D7D3C1E840D6C640C4C1E3C140C9D540C8C5E7C1C4C5C3C9D4C1D340D5D6
E3C1E3C9D6D540C9E240C3D6D5E3D9D6D3D3C5C440C2E840E3C8C540C8C5E7
-----
COMMAND.
C3D6D4D4C1D5C44B40
-----
THREE LINES ARE DISPLAYED FOR EACH SOURCE LINE.
```

Figure 34. Browse Hexadecimal Display - Data

LOCATE - Locating Lines

Use the LOCATE command to bring a particular line to the top of the display. You can identify the line by either its relative line number or a previously-defined label.

During Browse, the current position of the screen window is indicated by the line/column numbers in the upper-right corner of the screen. The line number refers to the first line of data following the two header lines, and shows the relative position of that line in the data. The **TOP OF DATA** message is treated as relative line zero.

The LOCATE command has the following format:

```
LOCATE {line-number | label}
```

You must enter either a line number or a label as an operand.

line-number

A numeric value of up to eight digits that shows the position of the line from the beginning of the data. The line number is displayed in the upper-right corner.

label Defined by scrolling to the top of the screen the line with which you want to associate the label. You then type the label on the COMMAND line in the form:

```
.ccccccc
```

For example, to find line 18463, you could enter the following command:

PDF then moves line 18463 to the top of the screen. You can assign a label to it by entering:

The label is a period followed by up to seven characters that can be displayed, except the comma and the space. It is treated as an internal symbol and is equated to the top line on the screen. You are required to specify the period when you define the label. The next time you want to find this line, you can enter:

The period is *usually* optional when you use it as an operand in a LOCATE command. However, if the first character in the label is a number, you must specify the period to distinguish the label from a line number.

The latest assignment of a label overrides any previous assignments. You can assign several labels to the same line. Labels are not retained when you leave the Browse option.

RESET - Removing the Column-Identification Line

The RESET command removes the column-identification line that you can display by using the COLUMNS command. This command has no operands. See "COLUMNS - Identifying Columns" on page 88 for information about the COLUMNS command. An example of the column-identification line is shown in Figure 32 on page 89.

Chapter 5. Edit (Option 2)

The Edit option (2) allows you to create, display, and change data stored in ISPF libraries or other partitioned or sequential data sets with the following characteristics:

- Record Format (RECFM):
 - Fixed or variable (non-spanned)
 - Blocked or unblocked
 - With or without printer control characters.
- Logical Record Length (LRECL):
 - From 10 to 255, inclusive, for fixed-length records
 - From 14 to 259, inclusive, for variable-length records.

Editing a Data Set

When you select the Edit option, the Edit - Entry Panel shown in Figure 35 is displayed.

```

----- EDIT - ENTRY PANEL -----
COMMAND ===>

ISPF LIBRARY:
PROJECT ===> ISPFDEMO
GROUP   ===> MYLIB   ===> MASTER   ===>
TYPE    ===> PLI
MEMBER  ===> _      (Blank or pattern for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ===>
VOLUME SERIAL ===> (If not cataloged)

DATA SET PASSWORD ===> (If password protected)

PROFILE NAME ===> (Blank defaults to data set type)

INITIAL MACRO ===> LOCK      ===> YES (YES, NO or NEVER)

FORMAT NAME   ===> MIXED MODE ===> NO (YES or NO)
  
```

Figure 35. Edit - Entry Panel

You can specify a concatenated sequence of up to four ISPF libraries, but the libraries must have been previously allocated to PDF with the Data Set utility (3.2).

The editor searches the ISPF libraries in the designated order to find the member and copy it into working storage. If you specified a nonexistent member of an ISPF library, a new member is created with the specified name.

When you save the edited member, the editor places or replaces it in the first ISPF library in the concatenation sequence, regardless of which library it was copied from. Figure 35 on page 99 shows the entries you would make to display the member list for the concatenated libraries ISPFDEMO.MYLIB.PLI and ISPFDEMO.MASTER.PLI.

Chapter 2, "Libraries and Data Sets" contains information about all the fields on the Edit - Entry Panel except:

PROFILE NAME

The Edit - Entry Panel allows you to specify a profile name, which you can enter to override the default Edit profile.

INITIAL MACRO

You can specify a macro to be executed before you begin editing your sequential data set or any member of a partitioned data set. This initial macro allows you to set up a particular editing environment for the Edit session you are beginning.

If you leave the INITIAL MACRO field blank and your edit profile includes an initial macro specification, the initial macro from your edit profile is executed.

If you want to suppress execution of an initial macro in your edit profile, enter NONE in the INITIAL MACRO field.

For complete information about the Edit option, see *Edit and Edit Macros*.

CHAPTER 6. UTILITIES (Option 3)

This chapter describes Utilities, option 3 on the ISPF/PDF Primary Option Menu, which provides a variety of functions. Among them are:

- **Maintaining libraries and data sets**
- **Moving and copying data**
- **Printing or displaying PDF project or TSO data set listings**
- **Resetting ISPF library statistics**
- **Initiating hardcopy output**
- **Retrieving VTOC entries for a DASD volume**
- **Retrieving held job output**
- **Creating and modifying application command tables**
- **Converting old format messages and menus to the new format**
- **Defining and maintaining formats for browsing and editing formatted data**
- **Comparing two or more data sets or members**
- **Searching for a character string.**

This chapter describes in detail each of these utilities.

Chapter 6. Utilities (Option 3)

The Utilities option (3) provides a variety of functions for library, data set, and catalog maintenance.

```

----- UTILITY SELECTION MENU -----
OPTION  ---> _

 1 LIBRARY   - Compress or print data set. Print index listing.
              Print, rename, delete, or browse members
 2 DATASET   - Allocate, rename, delete, catalog, uncatalog, or
              display information of an entire data set
 3 MOVE/COPY - Move, copy, or promote members or data sets
 4 DSLIST    - Print or display (to process) list of data set names
              Print or display VTOC information
 5 RESET     - Reset statistics for members of ISPF library
 6 HARDCOPY  - Initiate hardcopy output
 8 OUTLIST   - Display, delete, or print held job output
 9 COMMANDS  - Create/change an application command table
10 CONVERT   - Convert old format menus/messages to new format
11 FORMAT    - Format definition for formatted data Edit/Browse
12 SUPERC    - Compare data sets (Standard dialog)
13 SUPERCE   - Compare data sets (Extended dialog)
14 SEARCH-FOR - Search data sets for strings of data

```

Figure 36. Utility Selection Menu

The following sections describe the options shown on the Utility Selection Menu.

Library Utility (Option 3.1)

Library Utility (Option 3.1)

When you select this option, a panel is displayed that allows you to specify a data set and show an action to be done (Figure 37). The Library utility is intended primarily for maintenance of partitioned data sets. However, the print index listing (X), print entire data set (L), data set information (I), and data set information (short) (S) functions also apply to sequential data sets.

```
----- LIBRARY UTILITY -----
OPTION ---> _

blank - Display member list      B - Browse member
C - Compress data set           P - Print member
X - Print index listing         R - Rename member
L - Print entire data set       D - Delete member
I - Data set information        S - Data set information (short)

ISPF LIBRARY:
PROJECT ---> ISPFDEMO
GROUP ---> XXX ---> ---> --->
TYPE ---> COBOL
MEMBER --->
NEWNAME ---> (If "P", "R", "D", "B", or blank selected)
              (If "R" selected)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME --->
VOLUME SERIAL ---> (If not cataloged)
DATA SET PASSWORD ---> (If password protected)
```

Figure 37. Library Utility Panel

All the fields on the Library Utility panel, except one, are explained in Chapter 2, “Libraries and Data Sets.” That field is:

NEWNAME

You must specify a new member name in the NEWNAME field if you choose the R (rename member) option. See “R - Rename Member” on page 103 for more information about this option.

The following sections describe the options shown on the Library Utility panel.

Blank - Display Member List

If you leave the OPTION field blank, you must specify a partitioned data set. PDF displays a member list when you press the ENTER key. See “Using Member Selection Lists” on page 37 and “Library and Data Set List Utility Line Commands” on page 47 for more information.

B - Browse Member

If you select option B, you must specify a partitioned data set and a member name. The specified member is displayed in Browse mode. You can use all the Browse commands. When you end Browse by entering the END command, the Library Utility panel is displayed again.

C - Compress Data Set

If you select option C, you can specify any partitioned data set. The compress is accomplished by invoking either of the following:

- The IEBCOPY utility
- An optional compress request exit routine, which can be specified by your installation.

P - Print Member

If you select option P, you must specify a partitioned data set and a member name. A source listing of the member is recorded in the ISPF list data set.

Note: If any members are to be printed, the data set characteristics must conform to those for the L option.

X - Print Index Listing

If you select option X, you must specify either a DASD-resident sequential or partitioned data set. The index listing is recorded in the ISPF list data set. For a partitioned data set, the index listing includes general information about the data set followed by a member list. For a sequential data set, the index listing includes general information only. See Figure 186 on page 341 for an example of the index listing format.

Note: If PDF was entered in TEST mode, the listing also includes TTR data for each member of the data set. This data is the track and record address, where the members reside on the volume.

R - Rename Member

If you select option R, you must specify a partitioned data set and member name. You must also specify a new member name in the NEWNAME field.

Library Utility (Option 3.1)

L - Print Entire Data Set

If you select option L, you must specify either a DASD-resident sequential or partitioned data set. The allowable data set characteristics are the same as for Browse, except that data sets with a logical record length greater than 300 characters are not printed. Also, the data should not contain any printer control characters. Use the Hardcopy utility (3.6) to print data sets that contain printer control characters. A source listing of the complete data set (including all members of a partitioned data set), preceded by an index listing, is recorded in the ISPF list data set.

D - Delete Member

If you select option D, you must specify a partitioned data set and member name.

I - Data Set Information

If you select option I, the location, characteristics, and current space utilization of the specified data set are displayed (Figure 38). For sequential data sets, options I and S display the same information.

```
----- DATA SET INFORMATION -----
COMMAND ----> _

DATA SET NAME: ISPFDEMO.MYLIB.PLI

GENERAL DATA:                                CURRENT ALLOCATION:
Volume serial:          D00163                Allocated tracks:          3
Device type:            3350                  Allocated extents:         1
Organization:           PO                    Maximum dir. blocks:       30
Record format:          VB
Record length:          255
Block size:             6160
1st extent tracks:      3
Secondary tracks:       5

                                CURRENT UTILIZATION:
                                Used tracks:          3
                                Used extents:          1
                                Used dir. blocks:       1
                                Number of members:      1

Creation date:           1986/10/01
Expiration date:         ***NONE***
```

Figure 38. Library Utility - Data Set Information

S - Data Set Information (Short)

If you select option S, information about the selected data set is displayed. The information displayed by option S is the same information displayed by option S of the Data Set utility (3.2), but it differs from option I in two respects. Information for partitioned data sets, when displayed by option S, lacks the number of maximum and used directory blocks, and the number of members. For sequential data sets, options I and S display the same information. Figure 39 shows a short format example of data set information for a partitioned data set.

```
----- DATA SET INFORMATION -----
COMMAND ==> _

DATA SET NAME: ISPFDEMO.MYLIB.PLI

GENERAL DATA:                                CURRENT ALLOCATION:
  Volume serial:          D00163                Allocated tracks:          3
  Device type:            3350                  Allocated extents:           1
  Organization:          PO
  Record format:         VB
  Record length:         255
  Block size:            6160
  1st extent tracks:      3
  Secondary tracks:       5

                                CURRENT UTILIZATION:
                                Used tracks:          3
                                Used extents:          1

  Creation date:          1986/10/01
  Expiration date:        ***NONE***
```

Figure 39. Library Utility - Data Set Information (Short)

Data Set Utility (Option 3.2)

When you select this option, a panel is displayed that allows you to specify a data set and show an action to be done (Figure 40).

```
----- DATA SET UTILITY -----
OPTION ----> _

A - Allocate new data set          C - Catalog data set
R - Rename entire data set        U - Uncatalog data set
D - Delete entire data set        S - Data set information (short)
blank - Data set information

ISPF LIBRARY:
PROJECT ----> ISPFDEMO
GROUP  ----> MYLIB
TYPE   ----> PLI

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ---->
VOLUME SERIAL ---->          (If not cataloged, required for option "C")

DATA SET PASSWORD ---->      (If password protected)
```

Figure 40. Data Set Utility Panel

All the fields on the Data Set Utility panel are explained in Chapter 2, “Libraries and Data Sets.” For option A, you can specify any DASD-resident sequential or partitioned data set. For the other options, you can specify any DASD-resident data set that is not VSAM.

The following sections describe the options shown on the Data Set Utility panel.

A - Allocate New Data Set

Use option A to allocate a new data set. You must do the following:

- Enter one of the following:
 - An ISPF library name in the PROJECT, GROUP, and TYPE fields. See Chapter 2, “Libraries and Data Sets” if you need help entering the ISPF library name.
 - An “other” partitioned or sequential data set name in the DATA SET NAME field. See Chapter 2, “Libraries and Data Sets” if you need help entering the data set name.
- If you entered an ISPF library name, the value in the VOLUME SERIAL field is ignored.

However, if you entered an “other” data set name, you can specify the volume on which to allocate the data set in the VOLUME SERIAL field. If you do not specify a volume serial, PDF uses the system

catalog. See "Volume Serials" on page 32 if you need more information about volume serials.

- If your ISPF libraries and data sets are password-protected, enter the password in the DATA SET PASSWORD field. See "Using Data Set Passwords" on page 49 if you need more information about data set passwords.
- Press the ENTER key.

A panel is displayed to allow you to specify data set allocation values (Figure 41).

When you press the ENTER key with this panel displayed, the new data set is allocated and cataloged. Entering the END command returns you to the previous panel without allocating the data set.

An optional installation exit, the data set allocation exit, can control all data set creation, deletion, allocation, and deallocation done directly by PDF. This does not include allocations done by ISPF, the TSO ALLOCATE command, or other TSO commands. See *Installation and Customization* for more information about the data set allocation exit.

```

----- ALLOCATE NEW DATA SET -----
COMMAND ---->

DATA SET NAME: NEWPROJ.MASTER.ASM

VOLUME SERIAL      ----> -          (Blank for authorized default volume) *
GENERIC UNIT       ---->           (Generic group name or unit address) *
SPACE UNITS        ----> TRACK      (BLKS, TRKS, or CYLS)
PRIMARY QUANTITY   ----> 2          (In above units)
SECONDARY QUANTITY ----> 1          (In above units)
DIRECTORY BLOCKS   ----> 1          (Zero for sequential data set)
RECORD FORMAT      ----> FB
RECORD LENGTH      ----> 80
BLOCK SIZE         ----> 3120
EXPIRATION DATE    ----> 87/05/24   (YY/MM/DD,
                                     YY.DDD in Julian form,
                                     DDDD for retention period in days
                                     or blank)

( * Only one of these fields may be specified)

```

Figure 41. Allocate New Data Set Panel - Two-digit Year Support

Default values are provided for the fields on this panel, except for expiration date, based on which of the following occurred most recently:

- What you last entered on this panel
- The last display data set information request (option 3.1, 3.2, or 3.4).

Data Set Utility (Option 3.2)

You can type over the displayed defaults if you want to change them. The following is a list of the fields on this panel and their definitions:

VOLUME SERIAL

The volume serial of the direct access volume that is to contain the data set. This field overrides the VOLUME SERIAL field on the Data Set Utility panel. Therefore, you should leave this field blank if you want to do one of the following:

- Use the authorized default volume
- Enter a generic unit address in the GENERIC UNIT field.

GENERIC UNIT

The generic unit address for the direct access volume that is to contain the data set, such as 3330, 3350, 3380, and so forth. This field overrides the VOLUME SERIAL field on the Data Set Utility panel. Therefore, you should leave this field blank if you want to do one of the following:

- Use the authorized default volume
- Enter a volume serial in the VOLUME SERIAL field.

Notes:

1. *Leave both the VOLUME SERIAL and GENERIC UNIT fields blank to allow ISPF to select an eligible volume. Eligibility is determined by the "unit" information in your user entry in the TSO User Attribute Data Set (UADS).*
2. *At some installations, you are limited to eligible volumes (see note 1) even when an explicit volume serial is specified. At other installations you can specify any mounted volume. This is an installation option.*
3. *To allocate a data set to a 3850 virtual volume, you must also have MOUNT authority, gained by using the TSO ACCOUNT command.*

SPACE UNITS

Any of the following:

TRACK	Tells PDF that the amounts entered in the next two fields are expressed in tracks.
CYLINDER	Tells PDF that the amounts entered in the next two fields are expressed in cylinders.
BLOCK	Tells PDF that the amounts entered in the next two fields are expressed in blocks.

PRIMARY QUANTITY

The primary allocation quantity in tracks, cylinders, or blocks, as indicated in the SPACE UNITS field. This number can be zero for sequential data sets, but must be greater than zero for PDSs.

Also, if the primary quantity is zero, the secondary quantity must be greater than zero.

SECONDARY QUANTITY

The secondary allocation quantity in tracks, cylinders, or blocks, as indicated in the SPACE UNITS field. This quantity is allocated when the primary quantity is insufficient.

DIRECTORY BLOCKS

Enter one of the following:

- For *partitioned* data sets, you must specify the number of directory blocks. Each 256-byte block accommodates the following number of directory entries:
 - Data sets with ISPF statistics: 6
 - Data sets without ISPF statistics: 21
 - Load module data sets: 4 to 7, depending on attributes.
- For *sequential* data sets, set this field to zero.

RECORD FORMAT

Any valid combination of the following codes:

- | | |
|----------|--|
| F | Fixed-length records. |
| V | Variable-length records. |
| U | Undefined format records. |
| B | Blocked records. |
| A | ASA printer control characters. |
| M | Machine code printer control characters. |
| S | Standard (for F) or spanned (for V); use only with sequential data sets. |
| T | Track-overflow feature. |

Notes:

1. *You must enter either F, V, or U.*
2. *You can specify S and T, but ISPF does not otherwise support them.*

RECORD LENGTH

The logical record length, in bytes, of the records to be stored in the data set.

BLOCK SIZE

The block size, also called *physical* record length, of the blocks to be stored in the data set. Use this field to specify how many bytes of data to put into each block, based on the record length. For example, if the record length is 80 and the block size is 3120, 39 records can be placed in each block.

Note: The record length and block size are verified to be consistent with the record format, as defined in OS/VS2 MVS Data Management Services Guide. If you need to use non-standard characteristics, use the TSO ALLOCATE command.

EXPIRATION DATE

Lets you protect valuable data by specifying a date, in your national language, when the data set may be deleted. If you try to delete an unexpired data set, PDF displays two panels: a Confirm Delete panel, followed by a Confirm Purge panel. See "D - Delete Entire Data Set" on page 112 for more information about deleting unexpired data sets.

Four-digit year support is provided in the MVS/SP Version 2 Release 2 and MVS/XA DFP Version 2 Release 3 operating systems. If you have this support, you will see the panel shown in Figure 42 instead of the panel shown in Figure 41 on page 107.

```
----- ALLOCATE NEW DATA SET -----
COMMAND ==>

DATA SET NAME: NEWPROJ.MASTER.ASM

VOLUME SERIAL      ==> _          (Blank for authorized default volume) *
GENERIC UNIT       ==>           (Generic group name or unit address) *
SPACE UNITS        ==> TRACK      (BLKS, TRKS, or CYLS)
PRIMARY QUANTITY   ==> 2          (In above units)
SECONDARY QUANTITY ==> 1          (In above units)
DIRECTORY BLOCKS   ==> 1          (Zero for sequential data set)
RECORD FORMAT      ==> FB
RECORD LENGTH      ==> 80
BLOCK SIZE         ==> 3120
EXPIRATION DATE    ==> 1987/05/24 (YY/MM/DD, YYYY/MM/DD,
                                   YY.DDD, YYYY.DDD in Julian form,
                                   DDDD for retention period in days
                                   or blank)

( * Only one of these fields may be specified)
```

Figure 42. Allocate New Data Set Panel - Four-digit Year Support

An expiration date is not required, but if you enter one it should be in one of the following formats.

YY/MM/DD, YYYY/MM/DD

Date shown in year, month, and day, or your equivalent national format. The maximum expiration date allowed is:

- 2155/12/31 for a four-digit year
- 99/12/31 for a two-digit year.

YY.DDD, YYYY.DDD

Date shown in Julian format, such as 86.045 for February 14, 1986. The maximum expiration date allowed is:

- 2155.365 for a four-digit year
- Either 99.365 or 99.366 for a two-digit year.

DDDD The number of days, starting with the creation date, after which the data set can be deleted. DDDD has a range of 0 to 9999.

C - Catalog Data Set

If you select option C, the specified data set is cataloged. For this option, you must specify the volume serial on which the data set resides, regardless of whether the data set is specified as project, library, and type, or as another data set name. The data set must reside on the specified volume.

R - Rename Entire Data Set

If you select option R, a panel is displayed to allow you to enter the new data set name (Figure 43 on page 112). Type the new data set name and press the ENTER key to rename, or enter the END command to cancel. Either action returns you to the previous panel.

If you specify a volume serial for a data set to be renamed, PDF checks to see whether the data set is cataloged on that volume. If it is, the rename panel prompts you to specify whether to recatalog the data set. The displayed default for that field is YES. If you specify a volume serial and the data set is not cataloged, it remains uncataloged after you rename it. If a volume serial is not specified, the data set is recataloged to the new data set name and the old data set name is uncataloged.

If you are renaming a Generation Data Group (GDG) data set that has been defined with scratch and limit parameters, where the limit has been reached, you must specify a volume serial. The new data set name can then be cataloged with option C.

Also, you cannot catalog more than one version of the same generation of a Generation Data Group (GDG) data set. To rename a GDG data set that is cataloged, first give it a different generation number. Otherwise, the GDG data set is deleted.

Data Set Utility (Option 3.2)

```
----- RENAME DATA SET -----
COMMAND ---->

DATA SET NAME: ISPFDEMO.XXX.COBOL
VOLUME:      010901

ENTER NEW NAME BELOW:      (The data set will be recataloged.)

ISPF LIBRARY:
  PROJECT ---->
  GROUP  ---->
  TYPE   ---->

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
  DATA SET NAME ----> 'NEWPROJ.DEVEL.COBOL' _
```

Figure 43. Rename Data Set Panel

U - Uncatalog Data Set

If you select option U, the specified data set name is uncataloged. There is no need for the specified data set to be allocated or for the volume on which it resides to be mounted.

D - Delete Entire Data Set

If you select option D, a confirmation panel (Figure 44) is displayed so you can make sure you did not select this option by mistake.

```
----- CONFIRM DELETE -----
COMMAND ----> _

DATA SET NAME: ISPFDEMO.XXX.COBOL
VOLUME:      010901
CREATION DATE: 1986/11/19

INSTRUCTIONS:

  Press ENTER key to confirm delete request.
    (The data set will be deleted and uncataloged.)

  Enter END command to cancel delete request.
```

Figure 44. Confirm Delete Panel

If you specify a volume serial for the data set to be deleted, PDF checks to see whether the data set is cataloged on that volume. If so, the Confirm Delete panel prompts you to specify whether to uncatalog the data set. The displayed default is YES. If no volume serial is specified and the data set does not have an expiration date, the data set is deleted and uncataloged.

Note: PDF does not delete VSAM or password-protected data sets.

As directed on the panel, do one of the following:

- Press the ENTER key to confirm the data set deletion.
- Enter the END command to cancel. This action returns you to the previous panel.

If the data set has an expiration date that has not expired, PDF displays a Confirm Purge panel (Figure 45) after the Confirm Delete panel.

```
----- CONFIRM PURGE -----  
COMMAND ==>  
  
The data set being deleted has an expiration date which has not expired.  
  
DATA SET NAME:  ISPFDEMO.XXX.COBOL  
VOLUME:         010901  
CREATION DATE:  1986/11/19  
EXPIRATION DATE: 1999/12/31  
  
PURGE DATA SET ==> _ (YES or NO)  
  
INSTRUCTIONS:  
  
Enter YES to confirm the purge request.  
  (A request will be issued for the data set  
   to be deleted and uncataloged.)  
  
Enter NO or END command to cancel the purge request.
```

Figure 45. Confirm Purge Panel

This panel contains the following field:

PURGE DATA SET

Tells PDF whether to purge the data set. The valid values are:

YES Tells PDF to purge the data set.

NO Tells PDF *not* to purge the data set.

The statement that is enclosed in parentheses on the Confirm Purge panel shows whether the data set to be purged will be cataloged or uncataloged.

Data Set Utility (Option 3.2)

S - Data Set Information (Short)

If you select option S, information about the selected data set is displayed. The information displayed by option S is the same information displayed by option S on the Library Utility panel (option 3.1). See Figure 39 on page 105 for an example. To return to the previous panel, press the ENTER key or enter the END command.

Blank - Data Set Information

If you leave the OPTION field blank, information about the selected data set is displayed. The information displayed is the same information displayed by option I on the Library Utility panel (option 3.1). See Figure 38 on page 104 for an example. To return to the previous panel, press the ENTER key or enter the END command.

Move/Copy Utility (Option 3.3)

When you select this option, a panel is displayed that allows you to specify the “from” data set (and member if it is partitioned), and to select an option (Figure 46).

```

----- MOVE/COPY UTILITY -----
OPTION ==> C_

C - Copy data set or member(s)          CP - Copy and print
M - Move data set or member(s)         MP - Move and print
L - Copy and lock member(s)            LP - Copy, lock, and print
P - Promote data set or member(s)      PP - Promote and print

SPECIFY "FROM" DATA SET BELOW, THEN PRESS ENTER KEY

FROM ISPF LIBRARY:
PROJECT ==> ISPFDEMO
GROUP   ==> MYLIB
TYPE    ==> PLI
MEMBER  ==>                                     (Blank or pattern for member selection list,
                                                    '*' for all members)

FROM OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ==>
VOLUME SERIAL ==>                                     (If not cataloged)

DATA SET PASSWORD ==>                                     (If password protected)

```

Figure 46. Move/Copy Utility Panel

All the fields on the Move/Copy Utility panel are explained in Chapter 2, “Libraries and Data Sets.” On this panel, you specify the data set that you want to copy, move, lock, or promote. This is called the “from” data set.

If you request a member list or specify an asterisk (*) in the MEMBER NAME field on the “from” panel, PDF does not display a MEMBER NAME field on the “to” panel.

LMF library controls are required for the “from” library or data set when using options L and LP, and for the “target” library or data set when using the P and PP options. You can copy, move, lock, or promote load modules stored in partitioned data sets with “undefined” record formats, but you cannot print them.

The deletion of any member because of a move or promotion is recorded in your ISPF log data set, if allocated. For a promotion, the deletion is also recorded in the LMF activity log if an activity log is requested by the project administrator.

When you complete the panel and press the ENTER key, PDF displays another panel that is determined by the option you selected. This panel allows you to specify the “to” or “target” data set or controlled library. Both the “from” and “to” or “target” data sets must already exist. The utility does not automatically allocate space for a new “to” or “target” data set.

Move/Copy Utility (Option 3.3)

C and CP - Copying Data Sets

If you select C or CP, the panel shown in Figure 47 is displayed. This panel lets you specify the “to” data set—the library or data set name that you want the copied data stored under.

C - Copy Data Set or Member(s)

Use option C to copy a data set. You can specify either a DASD-resident sequential or partitioned data set for both the “from” or “to” data sets. The “from” data set is not deleted.

CP - Copy and Print

Same as option C, except source listings are recorded in the ISPF list data set, as follows:

- If the “to” data set is partitioned, a listing of each new or replaced member is recorded.
- If the “to” data set is sequential, a listing of its complete contents is recorded.

```
COPY --- FROM ISPFDEMO.MYLIB.PL1 -----
COMMAND ---->

SPECIFY "TO" DATA SET BELOW.

TO ISPF LIBRARY:
PROJECT ----> ISPFDEMO
GROUP  ----> MASTER_
TYPE   ----> PL1

TO OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ---->
VOLUME SERIAL  ---->          (If not cataloged)

DATA SET PASSWORD ---->          (If password protected)

"TO" DATA SET OPTIONS:
IF PARTITIONED, REPLACE LIKE-NAMED MEMBERS ----> YES   (YES or NO)
IF SEQUENTIAL, "TO" DATA SET DISPOSITION  ----> OLD   (OLD or MOD)
SPECIFY PACK OPTION FOR "TO" DATA SET      ----> NO    (YES, NO or blank)
```

Figure 47. Move/Copy Utility - “To” Panel for Copying

All the fields on the Move/Copy Utility “to” panels for copying data sets are explained in Chapter 2, “Libraries and Data Sets,” except the following:

IF PARTITIONED, REPLACE LIKE-NAMED MEMBERS

If the “to” data set is partitioned, enter:

- | | |
|------------|---|
| YES | To replace a member(s) in the “to” data set with a like-named member(s) in the “from” data set. |
| NO | To prevent a member(s) in the “to” data set from being replaced by a like-named member(s) in the “from” data set. |

IF SEQUENTIAL, “TO” DATA SET DISPOSITION

If the “to” data set is sequential, enter:

- | | |
|------------|--|
| OLD | To replace the “to” data set’s entire contents with the contents of the “from” data set. |
| MOD | To add the “from” data set to the end of the “to” data set. |

If the “from” data set consists of several members of an ISPF library or a partitioned data set to be moved or copied to a sequential data set, the members are written to the “to” data set one after another. The “to” data set disposition (OLD or MOD) controls only the beginning location of the “to” data set after the copy or move is completed.

SPECIFY PACK OPTION FOR “TO” DATA SET

Enter:

- | | |
|--------------|--|
| YES | If you want the data in the “to” data set to be packed. |
| NO | If you do not want the data in the “to” data set to be packed. |
| blank | If you leave this field blank, the data has the same format in the “to” data set as it had in the “from” data set. |

The technique used to pack data is the IEBCOPY program, used only by ISPF. If the data is packed, attempts to access or execute the data outside PDF can cause unwanted results. See the description of the PACK primary command in *Edit and Edit Macros* for more information.

Move/Copy Utility (Option 3.3)

M and MP - Moving Data Sets

If you select M or MP, the panel shown in Figure 48 is displayed. This panel lets you specify the “to” data set—the library or data set name that you want the moved data stored under.

M - Move Data Set or Member(s)

Use option M to move a data set. You can specify either a DASD-resident sequential or partitioned data set for both the “from” or “to” data sets.

Option M causes data sets to be deleted after they have been successfully moved to the “to” data set, as follows:

- If the “from” data set is partitioned, the selected members are deleted from it.
- If the “from” data set is sequential, the complete “from” data set is deleted.

MP - Move and Print

Same as option M, except source listings are recorded in the ISPF list data set, as follows:

- If the “to” data set is partitioned, a listing of each new or replaced member is recorded.
- If the “to” data set is sequential, a listing of its complete contents is recorded.

```
MOVE --- FROM ISPFDEMO.MYLIB.PL1 -----
COMMAND ---->

SPECIFY "TO" DATA SET BELOW.

TO ISPF LIBRARY:
PROJECT ----> ISPFDEMO
GROUP ----> MASTER_
TYPE ----> PL1

TO OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ---->
VOLUME SERIAL ----> (If not cataloged)

DATA SET PASSWORD ----> (If password protected)

"TO" DATA SET OPTIONS:
IF PARTITIONED, REPLACE LIKE-NAMED MEMBERS ----> YES (YES or NO)
IF SEQUENTIAL, "TO" DATA SET DISPOSITION ----> OLD (OLD or MOD)
SPECIFY PACK OPTION FOR "TO" DATA SET ----> NO (YES, NO or blank)
```

Figure 48. Move/Copy Utility - “To” Panel for Moving

All the fields on the Move/Copy Utility “to” panels for moving data sets are explained in Chapter 2, “Libraries and Data Sets,” except the following:

- IF PARTITIONED, REPLACE LIKE-NAMED MEMBERS
- IF SEQUENTIAL, “TO” DATA SET DISPOSITION
- SPECIFY PACK OPTION FOR “TO” DATA SET

See “C and CP - Copying Data Sets” on page 116 for descriptions of these fields.

L and LP - Copying and Locking Data Sets

If you select L or LP, the panel shown in Figure 47 on page 116 is displayed. This panel lets you specify the “to” data set—the library or data set name that you want the copied data set stored under.

The differences between using these options and using the C and CP options are:

- The “from” data set must be a controlled library member.
- L and LP lock the “from” member with your ID before copying it. No one else can change the member in the controlled library until you promote it.

The L and LP options are discussed thoroughly in *Library Management*. These options apply only to data sets controlled by the Library Management Facility (LMF).

L - Copy and Lock Member(s)

Use option L to copy data from a controlled library and lock the member in the controlled library. You can specify:

- A DASD-resident controlled library member for the “from” data set. The “from” data set is not deleted.
- A DASD-resident ISPF library or sequential data set for the “to” data set.

LP - Copy, Lock, and Print

Same as option L, except source listings are recorded in the ISPF list data set, as follows:

- If the “to” data set is partitioned, a listing of each new or replaced member is recorded.
- If the “to” data set is sequential, a listing of its complete contents is recorded.

Move/Copy Utility (Option 3.3)

P and PP - Promoting Data Sets

Select P or PP to do one of the following:

- Promote from a private library into an entry-level controlled library.
- Promote a member from an entry-level library into the next library in the hierarchy, as determined by the project administrator.
- Free a previously-locked member of a controlled library.

If the “from” library is a private library, once you enter P or PP and the “from” library or data set name, the panel shown in Figure 49 on page 121 is displayed. This panel lets you specify the “target” controlled library—the library that will receive the member.

If the “from” library is a controlled library, no “target” is displayed because the project administrator has already specified where the member is to be promoted.

Notes:

1. *The promote options apply only to data sets controlled by the Library Management Facility (LMF).*
2. *Promote exits can be invoked before and after members are promoted if the library administrator specified any exits when defining the library controls. These promote exits, along with the options and the promotion process, are discussed thoroughly in Library Management.*

P - Promote Data Set or Member(s)

Use option P to promote a data set into a controlled library. The “from” data set can be sequential, but the “target” data set must be a controlled ISPF library.

The selected members are deleted from the “from” data set after they have been successfully copied to the “to” data set.

PP - Promote and Print

Same as option P, except a source listing of each new or replaced member is recorded in the ISPF list data set.

```
PROMOTE --- FROM ISPFDEMO.MYLIB.PLI(ABCD) -----
COMMAND ---->

SPECIFY "TARGET" CONTROLLED LIBRARY BELOW.

TO ISPF LIBRARY:
PROJECT ----> ISPFDEMO
GROUP ----> MASTER
TYPE ----> PLI
MEMBER ----> _ (Blank unless member is to be renamed)

ACTION ----> MOVE (MOVE to copy and delete source
                  FREE to only delete source)

REASON CODE ---->

PACK DATA ----> (YES, NO, or blank to default to library controls)
```

Figure 49. Move/Copy Utility - "Target" Panel for Promoting

All the fields on the Move/Copy Utility "target" panel for promoting data sets are explained in Chapter 2, "Libraries and Data Sets" on page 29, except for the following:

ACTION

This field specifies whether a normal promotion of the member is to occur, or the member is to be made available for update. You can specify the latter action if you want to make the member available to others without promoting it. The specified actions are:

MOVE Promotion that moves the member to the target library and deletes it from the private library. The member becomes available for update, which means it is no longer locked by your logon ID.

FREE Makes the member available, no longer locked by your logon ID. The member is not moved to the target library, but is deleted from the private library.

REASON CODE

The information in this field will be written to the activity log. This is a 26-character field that you can use for tracking, testing in a user exit, or for any other purpose.

PACK DATA

You can use this field to override the library control. If the data is packed (or unpacked) when promoted into an entry-level library, it remains packed (or unpacked) throughout each later promotion.

Move/Copy Utility (Option 3.3)

Enter one of the following in the PACK DATA field:

- | | |
|--------------|---|
| YES | Tells PDF to pack (compress) the data set before promoting it. |
| NO | Tells PDF not to pack the data set before promoting it. |
| blank | Leaving this field blank defaults to the packed data library control specification. |

Using the Move/Copy Utility without Load Modules

For any move, copy, or promote operation that does not involve load modules, the following rules apply:

- Record formats (RECFM) and logical record lengths (LRECL) for the two data sets need not be alike. When necessary, data is truncated or padded on the right with blanks to accommodate different record lengths.
- For promote, if the record format is fixed, the logical record lengths must be the same, but the block size can be different. For promote with a variable record format, the target logical record length must be equal to or larger than the source logical record length. The blocks can differ, but must be logically consistent with the logical record length.
- The data to be moved, copied, or promoted is not renumbered or changed in any way, except for possible truncation, padding, or packing as noted above. Printer control characters, if present, are treated as data.
- Library statistics or other user data in directory entries are not changed when moving or copying between partitioned data sets.
- Partitioned data set members with user TTRs cannot be moved, copied, or promoted.

Using the Move/Copy Utility with Load Modules

For a move, copy, or promote of load modules, the following rules apply:

- Both data sets must be partitioned and must have an undefined record format (RECFM = U).
- No reblocking occurs. The block size of the “to” data set must be equal to or larger than the block size of the “from” data set. The logical record length (LRECL) is ignored.
- Load modules that were created for “planned overlay” cannot be moved or copied.

Moving, Copying, or Promoting Alias Entries

Alias entries can be moved or copied from one partitioned data set to another under the following conditions:

- When all members are specified (member name “*”) and the move or copy is completed successfully for the main member(s) to which the aliases refer.
- From a member list:
 - When the main member(s) are selected first, are not renamed, and are successfully moved or copied
 - The alias name(s) are then selected without leaving the list.

In all other cases for move and copy, and always for promote, you can select alias names but they are not preserved as aliases in the “to” data set; that is, the members to which they refer are moved or copied and the alias entries are stored in the “to” data set with the alias flags turned off.

Data Set List Utility (Option 3.4)

Data Set List Utility (Option 3.4)

When you select option 3.4, the Data Set List Utility panel (Figure 50) is displayed. You can either display or print lists of ISPF libraries, data sets, or volume table of contents (VTOC) information.

```
----- DATA SET LIST UTILITY -----
OPTION  ----> _

blank - Display data set list *      P - Print data set list
V    - Display VTOC information only PV - Print VTOC information only

Enter one or both of the parameters below:
DSNAME LEVEL  ----> USER
VOLUME        ---->

INITIAL DISPLAY VIEW  ----> VOLUME  (VOLUME,SPACE,ATTRIB,TOTAL)
CONFIRM DELETE REQUEST ----> YES    (YES or NO)

* The following line commands will be available when the list is displayed:

B - Browse data set      C - Catalog data set      F - Free unused space
E - Edit data set        U - Uncatalog data set     - - Repeat last command
D - Delete data set      P - Print data set
R - Rename data set      X - Print index listing
I - Data set information M - Display member list
S - Information (short)  Z - Compress data set      TSO command or CLIST
```

Figure 50. Data Set List Utility Panel

The fields on this panel are:

DSNAME LEVEL

This field is used to specify the level(s) of any data set that you want PDF to list or print for you. An optional installation exit, called the data set list (DSLISIT) exit, can control whether a data set name should appear in the list. See *Installation and Customization* for more information about this exit.

Note: The Data Set List utility (3.4) does not support tape or multi-volume data sets.

When you specify the DSNAME LEVEL, you are defining the level qualifiers for the data set names to be included in the list. Therefore, in the figure above, the value USER represents the first level of a data set name. For example, an ISPF library has a three-level name: project, group and type.

PDF retains the information you put in this field and displays it the next time you use this panel.

Except for the first level, you can specify the level qualifiers fully, partially, or use defaults. You must specify the first-level qualifier fully. Do not enclose the DSNAME LEVEL value in quotes.

Data Set List Utility (Option 3.4)

If you use an asterisk (*) to partially specify a qualifier, you must specify it as the first or last character in the qualifier. For example:

In this example, all data sets with ISPFDEVL as the first qualifier and SOURCE as the third qualifier are included in the list. There is no restriction on the second qualifier or any qualifiers beyond the third one. Here's another example:

In this example, all data sets with a first qualifier of SYS1 and with a second qualifier ending in LIB are included.

For options blank (Display data set list) and P (Print data set list), the following apply:

- If no matches are found and the data set name level does not contain an asterisk (*), the catalog or VTOC is searched for a data set name that matches the specified data set name level. If a match is found, a data set name is displayed or printed.
- A data set alias name is neither displayed nor printed. Instead, the data set name of which it is an alias is displayed or printed. This can cause a list to contain names that do not begin with the specified data set name level or duplicate entries.

VOLUME

Use this field to specify the volume serial whose VTOC is to be used by PDF to display or print a list of data set names or VTOC information. PDF retains the information you put in this field and displays it the next time you use this panel.

If you want to display a list of only the data sets that reside on a particular volume, leave the DSNAME LEVEL field blank and enter the volume serial in the VOLUME field.

INITIAL DISPLAY VIEW

Use this field to tell PDF which view of the data set list you would like to see. PDF retains the information you put in this field and displays it the next time you use this panel.

All the scroll commands are available from these displays and function normally, except for the LEFT and RIGHT commands. These commands switch from one view to another, because the panels used to show the different views are connected, as if they formed a "ring." Each time you enter the RIGHT command, another view is displayed in the sequence shown in Figure 51 on page 126, starting from the current view. Each time you enter the LEFT command, another view is displayed in *reverse* sequence.

Data Set List Utility (Option 3.4)

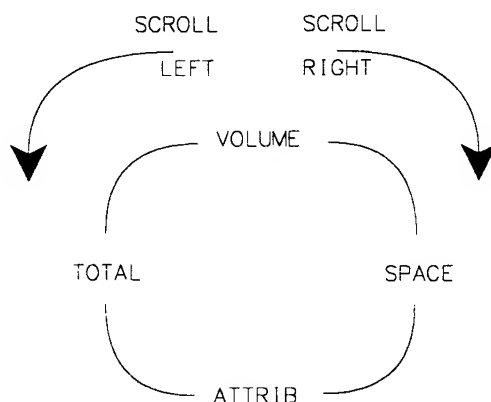


Figure 51. Sequence of Data Set List Display Views

If you enter the **RIGHT** command with the **TOTAL** view displayed, which is the last view in the sequence, PDF displays the **VOLUME** view, the first view in the sequence. If you enter the **LEFT** command with the **VOLUME** view displayed, PDF displays the **TOTAL** view. The available views are:

- | | |
|---------------|---|
| VOLUME | The VOLUME view shows a data set list that contains data set names and the volumes on which they reside. Figure 52 on page 128 shows a typical data set list display using the VOLUME view. The list is sorted by data set name. See the list under Figure 55 on page 129 for descriptions of the fields shown on this panel. |
| SPACE | The SPACE view shows a data set list that contains data set names, tracks, percentages used, extents, and devices. Figure 53 on page 128 shows a typical data set list display using the SPACE view. The list is sorted by data set name. See the list under Figure 55 on page 129 for descriptions of the fields shown on this panel. |
| ATTRIB | The ATTRIB view shows a data set list that contains data set names, data set organizations, record formats, logical record lengths, and block sizes. Figure 54 on page 129 shows a typical data set list display using the ATTRIB view. The list is sorted by data set name. See the list under Figure 55 on page 129 for descriptions of the fields shown on this panel. |
| TOTAL | The TOTAL view shows a data set list that contains all the information displayed by the VOLUME , SPACE , and ATTRIB views, plus the created, expired, and referred dates. Figure 55 on page 129 shows a typical data set list display using the TOTAL view. The list is sorted by data set name and has two lines per data set. See the list |

under the figure for descriptions of the fields shown on this panel.

CONFIRM DELETE REQUEST

This field controls whether the Confirm Delete panel is to be displayed when using the D (delete data set) line command from the displayed data set list or the TSO DELETE command. The two responses you can type in this field are:

YES A YES response tells PDF to display the Confirm Delete panel (Figure 44 on page 112), giving you an opportunity to change your mind and keep the data set. If you try to delete an unexpired data set, the Confirm Purge panel (Figure 45 on page 113) is displayed following the Confirm Delete panel. Follow the directions on the panel to either confirm or cancel the data set purge.

NO A NO response tells PDF not to display the Confirm Delete panel. The data set is deleted without your having to take any additional actions, unless you try to delete an unexpired data set. If this is the case, the Confirm Purge panel is displayed.

CAUTION

If you specify NO and the data set is deleted, it cannot be retrieved.

See “D - Delete Data Set” on page 138 for more information about the D line command.

The following sections describe the options shown on the Data Set List Utility panel.

Blank - Display Data Set List

Leave the OPTION field blank to display a data set list. Also, you must do the following:

- Enter one or more data set name level qualifiers in the DSNAME LEVEL field. See DSNAME LEVEL on page 124 if you need more information.
- Enter a volume serial in the VOLUME field if you want PDF to create a data set list from the VTOC. If you leave this field blank, the list is created from the catalog. See VOLUME on page 125 if you need more information.
- In the INITIAL DISPLAY VIEW field, enter the view of the data set list—VOLUME, SPACE, ATTRIB, or TOTAL—that you want to see first. Examples of these views are shown in Figure 52 on page 128, Figure 53 on page 128, Figure 54 on page 129, and Figure 55 on page 129, respectively.

Data Set List Utility (Option 3.4)

- Enter YES or NO in the CONFIRM DELETE REQUEST field to tell PDF whether to display a confirmation panel if you enter the D (delete data set) line command or the TSO DELETE command. See CONFIRM DELETE REQUEST on page 127 if you need more information.
- Press the ENTER key to display the data set list.

DSLST - DATA SETS BEGINNING WITH USER ----- ROW 1 OF 17
COMMAND ----> _ SCROLL ----> PAGE

COMMAND	NAME	MESSAGE	VOLUME
	USER.AVERAGE.ASM		TSOPK1
	USER.AVERAGE.LOAD		TSOPK6
	USER.AVERAGE.OBJ		TSOPK6
	USER.AVERAGE.TEXT		TSOPK3
	USER.CLIST		TSOPK2
	USER.CLISTS		TSOPK6
	USER.COMPARE.LISTING		TSOPK6
	USER.CUT.CLIST		TSOPK6
	USER.DATA.FILE		TSOPK4
	USER.DBA2P010.CLIST		TSOPK4
	USER.DBA2T010.CLIST		TSOPK4
	USER.ISAPISPF.CLIST		TSOPK4
	USER.SMITH.CLIST		TSOPK4
	USER.SMITH.LOAD		TSOPK4
	USER.SMITH.MSGS		TSOPK4
	USER.SMITH.PANELS		TSOPK4
	USER.SMITH.SKELS		TSOPK4
***** END OF DATA SET LIST *****			

Figure 52. Data Set List - VOLUME View

DSLST - DATA SETS BEGINNING WITH USER ----- ROW 1 OF 17
COMMAND ----> _ SCROLL ----> PAGE

COMMAND	NAME	TRACKS	%USED	XT	DEVICE
	USER.AVERAGE.ASM	5	20	1	3330
	USER.AVERAGE.LOAD	5	20	1	3380
	USER.AVERAGE.OBJ	5	20	1	3380
	USER.AVERAGE.TEXT	350	100	2	3380
	USER.CLIST	55	40	1	3380
	USER.CLISTS	65	30	1	3380
	USER.COMPARE.LISTING	25	100	1	3380
	USER.CUT.CLIST	1	100	6	3380
	USER.DATA.FILE	2	78	1	3350
	USER.DBA2P010.CLIST	15	95	1	3350
	USER.DBA2T010.CLIST	10	97	1	3350
	USER.ISAPISPF.CLIST	66	73	7	3350
	USER.SMITH.CLIST	5	50	1	3350
	USER.SMITH.LOAD	15	40	1	3350
	USER.SMITH.MSGS	5	95	1	3350
	USER.SMITH.PANELS	5	95	1	3350
	USER.SMITH.SKELS	5	60	1	3350
***** END OF DATA SET LIST *****					

Figure 53. Data Set List - SPACE View

Data Set List Utility (Option 3.4)

DSLST - DATA SETS BEGINNING WITH USER -----						ROW 1 OF 17
COMMAND ----> _						SCROLL ----> PAGE
COMMAND	NAME	DSORG	RECFM	LRECL	BLKSZ	
	USER.AVERAGE.TEXT	PS	FB	80	12960	
	USER.AVERAGE.ASM	PS	FB	80	6120	
	USER.AVERAGE.LOAD	PO	U	0	6144	
	USER.AVERAGE.OBJ	PS	FB	80	80	
	USER.CLIST	PO	VB	255	6160	
	USER.CLISTS	PO	VB	255	6160	
	USER.COMPARE.LISTING	PO	FBA	133	12901	
	USER.CUT.CLIST	PS	FB	80	12960	
	USER.DATA.FILE	PS	FB	200	12800	
	USER.DBA2P010.CLIST	PO	FB	80	12960	
	USER.DBA2T010.CLIST	PO	FB	80	12960	
	USER.ISAPISPF.CLIST	PO	FB	80	12960	
	USER.SMITH.CLIST	PO	FB	80	12960	
	USER.SMITH.LOAD	PO	U	0	6144	
	USER.SMITH.MSGS	PO	FB	80	12960	
	USER.SMITH.PANELS	PO	FB	80	12960	
	USER.SMITH.SKELS	PO	FB	80	12960	
***** END OF DATA SET LIST *****						

Figure 54. Data Set List - ATTRIB View

DSLST - DATA SETS BEGINNING WITH USER -----										ROW 1 OF 17
COMMAND ----> _										SCROLL ----> PAGE
COMMAND	NAME					MESSAGE		VOLUME		
TRACKS %	XT	DEVICE	DSORG	RECFM	LRECL	BLKSZ	CREATED	EXPIRES	REFERRED	
USER.AVERAGE.ASM										TSOPK1
5 20	1	3330	PS	FB	80	6120	1986/01/05	1988/01/05	1986/01/12	
USER.AVERAGE.LOAD										TSOPK6
5 20	1	3380	PO	U	0	6144	1986/01/05	1988/01/05	1986/01/12	
USER.AVERAGE.OBJ										TSOPK6
5 20	1	3380	PS	FB	80	80	1986/01/05	1988/01/05	1986/01/12	
USER.AVERAGE.TEXT										TSOPK3
350 100	2	3380	PS	FB	80	12960	1985/11/11	***NONE***	1985/11/11	
USER.CLIST										TSOPK2
55 40	1	3380	PO	VB	255	6160	1986/01/05	1988/01/05	1986/01/12	
USER.CLISTS										TSOPK6
65 30	1	3380	PO	VB	255	6160	1986/01/05	1988/01/05	1986/01/12	

Figure 55. Data Set List - TOTAL View

The fields on the display view panels are:

COMMAND Field used to enter a line command, TSO command, or CLIST when displaying a data set list. See "Blank - Display Data Set List" on page 127 if you need more information.

NAME Data set name, as in the VTOC.

MESSAGE This field is initially blank. However, after you carry out an operation on a data set by using one of the built-in line commands listed on the Data Set List Utility panel, a message appears in this field showing the last function

Data Set List Utility (Option 3.4)

used on that data set. The line commands and the messages that appear when they are used are:

Line Command	Message	Line Command	Message
B	BROWSED	C	CATALOGED
E	EDITED	U	UNCATALOGED
D	DELETED	P	PRINTED
R	RENAMED	X	INDEX PRINTED
I	INFO-I	M	MEMBER LIST
S	INFO-S	Z	COMPRESSED
F	SPACE FREED	=	Message shown for last command entered.

Figure 56. Data Set List Utility Line Commands and Messages

If you enter a TSO command or CLIST in the COMMAND field, a default message appears in the MESSAGE field. The message you see depends on the results of the TSO command or CLIST, and has the following format:

XXXXXXXX RC=#

where:

XXXXXXXX The command entered.

The return code.

Note: See "Data Set List Utility Line Commands" on page 138 for a description of the Data Set List Utility line commands.

VOLUME Volume serial number.

TRACKS Number of tracks allocated to the data set.

% Percentage of allocated tracks used, expressed in whole numbers, not rounded. If any track is used, the minimum percentage is 1.

Note: See "F - Free Unused Space" on page 141 for information about freeing track space manually.

XT Number of extents allocated to the data set.

This option processes data by volume, and a data set can contain a maximum of 16 extents. Therefore, if a data set, such as a VSAM data set, contains more than 16 extents, ISPF cannot determine the number of tracks, so it substitutes ??? for this number.

Data Set List Utility (Option 3.4)

DEVICE	Device type on which the volume that contains the data set is mounted.																				
DSORG	One of the following data set organizations. In the definitions of these data set organizations, <i>unmovable</i> means the data set contains absolute addresses instead of relative addresses. These data sets are not moved to any other DASD storage location during read/write operations. <table><tr><td>PS</td><td>Sequential.</td></tr><tr><td>PSU</td><td>Sequential unmovable.</td></tr><tr><td>PO</td><td>Partitioned.</td></tr><tr><td>POU</td><td>Partitioned unmovable.</td></tr><tr><td>DA</td><td>Direct.</td></tr><tr><td>DAU</td><td>Direct unmovable.</td></tr><tr><td>IS</td><td>Indexed sequential.</td></tr><tr><td>ISU</td><td>Indexed sequential unmovable.</td></tr><tr><td>VS</td><td>VSAM.</td></tr><tr><td>blank</td><td>None of the above data set organizations.</td></tr></table>	PS	Sequential.	PSU	Sequential unmovable.	PO	Partitioned.	POU	Partitioned unmovable.	DA	Direct.	DAU	Direct unmovable.	IS	Indexed sequential.	ISU	Indexed sequential unmovable.	VS	VSAM.	blank	None of the above data set organizations.
PS	Sequential.																				
PSU	Sequential unmovable.																				
PO	Partitioned.																				
POU	Partitioned unmovable.																				
DA	Direct.																				
DAU	Direct unmovable.																				
IS	Indexed sequential.																				
ISU	Indexed sequential unmovable.																				
VS	VSAM.																				
blank	None of the above data set organizations.																				
RECFM	Record format specified when the data set was allocated. See "A - Allocate New Data Set" on page 106 if you need more information about record formats.																				
LRECL	Logical record length, in bytes, specified when the data set was allocated.																				
BLKSZ	Block size, in bytes, specified when the data set was allocated.																				
CREATED	Creation date in the national format.																				
EXPIRES	Expiration date in the national format, specified when the data set was allocated. If no expiration date was specified, ***NONE*** is displayed. See "A - Allocate New Data Set" on page 106 if you need more information about expiration dates.																				
REFERRED	Date, in the national format, that this data set was last accessed.																				

Line Commands

The asterisk (*) that appears after this option refers to the bottom part of the panel, which contains a list of line commands that are available when you display a data set list. These line commands can be entered in the Command field to the left of the data set names. See "Data Set List Utility Line Commands" on page 138 for definitions of these line commands.

Data Set List Utility (Option 3.4)

TSO Commands and CLISTs

Besides the PDF-supplied line commands, you can also enter TSO commands and CLISTs that use a fully-qualified data set name as a parameter. You can type over the NAME field to enter commands that require more space than is provided in the COMMAND field. PDF determines the end of the command by scanning the COMMAND and NAME fields from right to left. The first character found that differs from the original is considered to be the last character of the command. Therefore, it is best to enter a blank after the last character of your command if it extends into the NAME field.

Note: If the TSO command or CLIST issues a return code greater than or equal to eight, processing stops and an error message is displayed.

Using the Slash (/) Symbol

If a command or CLIST requires the data set name in a position other than the first parameter or if other parameters are needed, you can use the “/” symbol to represent the quoted data set name. If no arguments are specified after the command, PDF uses the name of the data set being acted on as the command’s first parameter.

To specify a member of a partitioned data set, enclose the member name in parentheses immediately following the “/” symbol. You can use this format with the B (browse data set), E (edit data set), and M (display member list) line commands. For information about these line commands, see “Data Set List Utility Line Commands” on page 138.

You may find it helpful to invoke the SHOWCMD primary command before using the “/” for the first time. After you invoke SHOWCMD, a special Data Set List Utility panel appears each time you enter a line command, TSO command, or CLIST on a data set list display. The panel shows you the command you entered and how PDF expanded, and thus interpreted, that command. See “SHOWCMD Command” on page 144 for more information about and an example of the SHOWCMD primary command.

The following rules apply to substituting the “/” character for a data set name. Each rule is followed by one or more examples that prove the rule by using either a CLIST or a line command.

In each example, the data set being acted on is USER.TEST.DATA, which always appears, either completely or partially, in uppercase. However, the CLIST or line command is typed in lowercase to differentiate between the CLIST or line command and USER.TEST.DATA when this data set name is either completely or partially typed over.

Each example also shows:

Original The line as it appears before the CLIST or line command is entered.

As typed The line as it appears after the CLIST or line command is typed.

Data Set List Utility (Option 3.4)

After The line as it appears after the CLIST or line command is expanded to show the placement of quotes and data set name substitution for the “/” symbol.

1. You can type over the data set name. Expanded commands can contain a maximum of 255 characters and are translated to uppercase. The following example shows how rule 1 would apply if you entered %CLIST1 DA(/):

```
(Original)          USER.TEST.DATA
(As typed)          %CLIST1 DA( / ) .TEST.DATA
(After)             %CLIST1 DA('USER.TEST.DATA')
```

2. The data set name substitution character (/) is replaced with the quoted, fully-qualified data set name if the character following the “/” is *not* a number, letter, or national character. The following example shows how rule 2 would apply if you entered %CLIST2 / NEWDATE(1986/03/15):

```
(Original)          USER.TEST.DATA
(As typed)          %CLIST2 / NEWDATE 1986 03 15
(After)             %CLIST2 'USER.TEST.DATA' NEWDATE(1986/03/15)
```

3. If a “/” is followed immediately by a member name in parentheses, the ending quote for the data set is placed after the closing parenthesis that follows the member name. The following example shows how rule 3 would apply if you entered %CLIST3 DA(/(XYZ)):

```
(Original)          USER.TEST.DATA
(As typed)          %CLIST3 DA( / (XYZ) ) .DATA
(After)             %CLIST3 DA('USER.TEST.DATA(XYZ)')
```

4. If the first argument is the unquoted data set name as it appears in the list, quotes are added around it or after a closing parenthesis following a member name. The following example shows how rule 4 would apply if you entered the B (browse) line command:

```
(Original)          USER.TEST.DATA
(As typed)          B USER.TEST.DATA(ABC)
(After)             B 'USER.TEST.DATA(ABC)'
```

5. If the line command does not have any arguments or if the data set name has not been substituted as specified by either rule 3 or rule 4, the quoted, fully-qualified data set name is added to the end of the line command. The following example shows how rule 5 would apply if you entered %CLIST4 USER.TEST.FORTRAN:

```
(Original)          USER.TEST.DATA
(As typed)          %CLIST4 user.test.fortran
(After)             %CLIST4 USER.TEST.FORTRAN 'USER.TEST.DATA'
```

Data Set List Utility (Option 3.4)

The following example shows how rule 5 on page 133 would apply if you entered %CLIST4 'USER.TEST.FORTRAN'. The purpose of this example is to show that if you enclose the CLIST argument in quotes, PDF still puts quotes around the data set name being acted on. The results are the same.

```
(Original)          USER.TEST.DATA
(As typed)
(After)      %CLIST4 'USER.TEST.FORTRAN' 'USER.TEST.DATA'
```

The following example shows how rule 5 on page 133 would apply if you entered %CLIST5 MEMBER1(ABC). The purpose of this example is to show that the results do not change if the CLIST argument contains a member name enclosed in parentheses.

```
(Original)          USER.TEST.DATA
(As typed)          ATA
(After)      %CLIST5 MEMBER1(ABC) 'USER.TEST.DATA'
```

TSO Command/CLIST Variables

If you use a TSO command or CLIST, PDF puts the following variables in the shared pool for the TSO command or CLIST to use:

Variable Name	Description	Length in Characters
ZDLDSN	Data set name	(44)
ZDLVOL	Volume	(6)
ZDLDEV	Device type	(8)
ZDLDSORG	Data set organization	(4)
ZDLRECFM	Data set record format	(5)
ZDLLRECL	Data set logical record length	(5)
ZDLBLKSZ	Data set block size	(5)
ZDLSIZE	Data set size in tracks	(6)
ZDLUSED	Percentage of used tracks	(3)
ZDLEXT	Number of extents used	(3)
ZDLCDATE	Creation date	(10)
ZDLEDATE	Expiration date	(10)
ZDLRDATE	Date last referenced	(10)
ZDLCONF	Delete confirmation (Y N)	(1)
ZDLLCMD	Expanded line command	(255)

Figure 57 (Part 1 of 2). TSO Command/CLIST Variables (Output)

Variable Name	Description	Length in Characters
ZDLCAT	Cataloged status; one of the following: <div> <div>0</div> <div>Data set is cataloged on volume ZDLVOL.</div> </div> <div> <div>2</div> <div>Data set is cataloged on a volume other than ZDLVOL and is uncataloged on volume ZDLVOL.</div> </div> <div> <div>4</div> <div>Data set is uncataloged on volume ZDLVOL.</div> </div> <div> <div>6</div> <div>Data set is not cataloged on any volume and is uncataloged on volume ZDLVOL.</div> </div> <div> <div>8</div> <div>Data set is not available on volume ZDLVOL. This status is returned for data sets that have been either migrated or deleted.</div> </div>	(1)

Figure 57 (Part 2 of 2). TSO Command/CLIST Variables (Output)

A TSO command or CLIST can set the following variables and place them in the shared pool to communicate with the Data Set List utility (3.4).

Variable Name	Description	Length in Characters
ZDLNDSN	New data set name to appear in list	(44)
ZDLMSG	Message to appear in list	(16)
ZDLREF	Refresh data set information; Y N	(1)

Figure 58. TSO Command/CLIST Variables (Input)

P - Print Data Set List

Use option P to print a data set list. You must do the following:

- Enter one of the following:
 - One or more data set name level qualifiers in the DSNAME LEVEL field and a volume serial in the VOLUME field. The list will contain all data sets for the specified level(s) and volume. Only the specified volume is searched. See DSNAME LEVEL on page 124 and VOLUME on page 125 if you need more information.
 - One or more data set name level qualifiers in the DSNAME LEVEL field, but leave the VOLUME field blank. The list will contain all data sets for the specified level(s). Every volume is searched.

Data Set List Utility (Option 3.4)

- A volume serial in the VOLUME field, but leave the DSNAME LEVEL field blank. The list will contain only the data sets on the specified volume. Only the specified volume is searched.

Note: All data set lists are formatted the same when they are printed. Therefore, values entered in the INITIAL DISPLAY VIEW field have no effect when you use option P. Figure 193 on page 347 shows the format of a printed data set list.

- Press the ENTER key to print the data set list. The data set list is stored in the ISPF list data set.

V - Display VTOC Information

Use option V to display VTOC (volume table of contents) information. The List VTOC utility (formerly option 3.7) has been removed from PDF. Options V and PV do the same function formerly provided by this utility.

To use option V, you must do the following:

- Blank out the DSNAME LEVEL field.
- In the VOLUME field, specify the volume serial for which you want PDF to display information.

Note: VTOC information is formatted the same, whether displayed or printed. Therefore, values entered in the INITIAL DISPLAY VIEW field have no effect when you use option V.

- Press the ENTER key to display the VTOC information.

Figure 59 shows an example of a VTOC display.

VTOC SUMMARY INFORMATION FOR VOLUME TSOPKA -----					
COMMAND ===> _					
UNIT: 3350					
VOLUME DATA:		VTOC DATA		FREE SPACE:	TRACKS
TRACKS:	16,800	TRACKS:	30	SIZE:	3,282
%USED:	80	%USED:	26	LARGEST:	1,140
TRKS/CYLS:	30	FREE DSCBS:	1,057	FREE EXTENTS:	129
					CYLS
					86
					38

Figure 59. VTOC Information Display

Track values do not include the remaining alternate tracks for the volume. The free space track values are the number of tracks for the free cylinders plus any additional free tracks.

The fields shown on the VTOC display are:

UNIT Shows the type of DASD device the volume is on, such as 3330, 3350, 3380, and so forth.

VOLUME DATA

Describes general information about the volume:

TRACKS Total tracks on the volume.

%USED Percentage of total tracks not available for allocation.

TRKS/CYLS Number of tracks per cylinder for this volume.

VTOC DATA

Describes general information about the VTOC on the volume:

TRACKS Total tracks allocated to the VTOC.

%USED Percentage of allocated tracks used by data set control blocks (DSCBs).

FREE DSCBS Number of unused DSCBs.

FREE SPACE

Describes the free space available for data set allocation on the volume under the headings **TRACKS** and **CYLS**, showing:

SIZE Total number of free tracks and cylinders.

LARGEST

The largest number of contiguous free tracks and cylinders.

FREE EXTENTS

The number of free areas with free cylinders.

PV - Print VTOC Information

Use option PV to print VTOC information. The List VTOC utility (formerly option 3.7) has been removed from PDF. Options V and PV do the same function formerly provided by this utility.

To use option PV, you must do the following:

- Blank out the DSNAME LEVEL field.
- In the VOLUME field, specify the volume serial for which you want to print information.

Data Set List Utility (Option 3.4)

Note: VTOC information is formatted the same, whether displayed or printed. Therefore, values entered in the INITIAL DISPLAY VIEW field have no effect when using option PV.

- Press the ENTER key to print the VTOC information. The VTOC information is stored in the ISPF list data set.

Data Set List Utility Line Commands

After you display a data set list by leaving the OPTION field blank, you can enter a line command to the left of the data set name. You can also enter TSO commands or CLIST names. The “/” symbol, which can be used with TSO commands and CLISTs, can also be used with the B (Browse data set), E (Edit data set), and M (Display member list) line commands to specify a member name or a pattern, such as /(MEM1) or /(MEM*), respectively. You can type over the Data Set Name field to enter commands that require more than the space provided. For more information about using this symbol, see “Using the Slash (/) Symbol” on page 132.

The available line commands are shown at the bottom of the Data Set List Utility panel and described below:

B - Browse Data Set

Processing is the same as with the Browse option (1), except that mixed mode is the assumed operation mode and you cannot specify a data set format. To browse a data set with mixed mode off, you must use the Browse option and enter NO in the MIXED MODE field on the Browse - Entry Panel. If you enter the B line command beside an ISPF library or other partitioned data set, a Browse member list is displayed. See “Using Member Selection Lists” on page 37 for more information.

E - Edit Data Set

For the E command, the processing is the same as if you had selected the Edit option (2) and entered the library or data set name on the Edit - Entry Panel, except that mixed mode is the assumed operation mode and you cannot specify a data set format, an edit profile, or an initial macro. If you select a library or other partitioned data set, an Edit member list is displayed. See “Using Member Selection Lists” on page 37 for more information.

D - Delete Data Set

For the D command, the processing is the same as if you had selected option D from the Data Set utility (3.2). This command deletes the complete data set.

If you specify YES in the CONFIRM DELETE REQUEST field on the Data Set List Utility panel, the Confirm Delete panel (Figure 44 on page 112) is displayed to allow you to continue or cancel the operation. If you are trying to delete an unexpired data set, the Confirm Purge panel (Figure 45 on page 113) is also displayed.

R - Rename Data Set

For the R command, the processing is the same as if you had selected option R from the Data Set utility (3.2). The Rename Data Set panel (Figure 43 on page 112) is displayed to let you specify the new name.

I - Data Set Information

For the I command, the processing is the same as if you had selected option I from the Library utility (3.1) or left the OPTION field blank with the Data Set utility (3.2).

S - Information (Short)

For the S command, the processing is the same as if you had selected option S from the Library utility (3.1) or the Data Set utility (3.2).

C - Catalog Data Set

For the C command, the processing is the same as if you had selected option C from the Data Set utility (3.2).

U - Uncatalog Data Set

For the U command, the processing is the same as if you had selected option U from the Data Set utility (3.2).

P - Print Data Set

For the P command, the processing is the same as if you had selected option L from the Library utility (3.1). This command formats the contents of a source data set for printing and records the output in the ISPF list data set. It also produces an index listing, which appears at the beginning of the output. Figure 185 on page 340 shows a sample source listing.

X - Print Index Listing

For the X command, the processing is the same as if you had selected option X from the Library utility (3.1). An index listing shows general data set information and member names for partitioned data sets. For ISPF source libraries, activity statistics are listed for each member. Figure 186 on page 341 shows a sample index listing for an ISPF library. For load libraries, load module information is listed for each member. Figure 187 on page 342 shows a sample index listing for a load library. The index listing is recorded in the ISPF list data set.

M - Display Member List

The M command displays a member selection list of a partitioned data set. This member list provides an expanded line command field in the area to the left of the list. The line command field in other member lists has room for only one character.

NW: 15457 DocId: 70002077 Page 326 From the member list, you can use the same primary commands and line commands that are valid for Library utility (3.1) member selection lists, plus

Data Set List Utility (Option 3.4)

the E (Edit data set member) line command. See “Using Member Selection Lists” on page 37 and “Library and Data Set List Utility Line Commands” on page 47 if you need more information.

You can also enter TSO commands or CLISTs in the line command field. If you enter a line command other than B (browse), D (delete), E (edit), R (rename), or P (print), PDF interprets it as a TSO command or CLIST. The command is selected with the quoted, fully-qualified data set name and member added to it, such as:

```
COMMAND 'DSN(MEMBER)'
```

See “TSO Commands and CLISTs” on page 132 if you need more information.

Note: If the TSO command or CLIST issues a return code greater than or equal to eight, processing stops and an error message is displayed.

Figure 60 shows an example of a member list with statistics and an expanded line command field. This panel contains the same information as Figure 12 on page 40, except the “CREATED” field is deleted.

DSLIST -- FBROWN.PRIVATE.PANELS -----									
COMMAND ---->									
NAME RENAME LIB VV.MM CHANGED SIZE INIT MOD ID									

ISRB00		3	21.11	83/12/05 09:12	29	23	0	DONLEY	
ISRB01		2	20.04	85/12/13 08:51	29	23	0	MARTIN	
ISRB10		3	21.00	84/03/26 10:53	29	29	0	PATTEN	
ISRB15		3	20.03	83/12/05 09:12	26	23	0	BROWN	
ISRB16		3	20.01	82/11/05 09:24	20	20	0	KISH	
ISRB18		2	21.01	86/01/21 14:03	17	11	0	BERNACK	
ISRB50		3	21.00	84/03/26 10:53	24	24	0	WALTERS	
ISRC00		1		ALIAS					
ISRC60		1	21.02	83/11/10 11:48	11	8	0	ROGERS	
ISRE00		3	21.01	83/02/04 11:05	26	26	0	SPENCER	
ISRE01		3	21.04	83/07/18 09:58	29	29	0	RICHEY	
ISRE02		3	21.11	83/07/18 10:00	15	8	0	GURLEY	
ISRE03		3	21.04	83/07/18 10:02	29	26	0	DUFF	
ISRE04		3	20.16	83/06/22 11:03	29	44	0	COONS	
ISRE05		2	01.01	85/10/04 11:13	29	29	0	RUPKAL	
ISRE06		3	20.10	84/02/17 14:31	29	29	0	MANNINO	
ISRE07		3	20.02	85/06/17 14:56	29	29	0	KELLRMA	
ISRE08		2	20.01	86/01/20 09:04	23	20	0	MCINDOO	
ISRE09		2	21.13	85/12/13 08:52	29	1	0	DIRKSE	
ISRE10		3	20.03	83/12/05 09:13	29	29	0	SEABOLD	
END									

Figure 60. Member List Display - Expanded Line Command Field

Figure 61 on page 141 shows load module library statistics with an expanded line command field. This panel contains the same information shown in Figure 13 on page 41, except the “ALIAS-OF” field is deleted.

Data Set List Utility (Option 3.4)

```

LIBRARY - FBROWN.PRIVATE.LOAD ----- ROW 00001 OF 00021
COMMAND ---->----- SCROLL ----> CSR

```

NAME	RENAME	LIB	SIZE	TTR	AC	----- ATTRIBUTES -----
IGX00036		2	0010F0	000D06	00 FO	RN RU
ISPF		2	000440	000313	00 FO	RN RU
ISRBRO		2	000BD8	000F09	00 FO	RN RU
ISRECMBR		2	000138	000025	00 FO	RN RU
ISREDIT		2	000EC8	000F23	00 FO	RN RU
ISRFMT		2	0028F0	000A06	00 FO	RN RU
ISRFPR		2	001290	000F01	00 FO	RN RU
ISRFPT		2	000278	00002D	00 FO	RN RU
ISRGE77		2	000768	000034	00 FO	RN RU
ISRGE77A		2	000768	000106	00 FO	RN RU
ISRGE78		2	000768	00010E	00 FO	RN RU
ISRGE78A		2	000768	000116	00 FO	RN RU
ISRJB1		2	000B40	00011E	00 FO	RN RU
ISRJB2		2	0012F8	000F11	00 FO	RN RU
ISRKA78K		2	000768	000126	00 FO	RN RU
ISRLALIN		2	007210	000A11	00 FO	RN RU
ISRLCFCL		2	011B00	000738	00 FO	RN RU
ISRLEMV		2	009048	000D0E	00 FO	RN RU
ISRLEQDS		2	0006B8	00012E	00 FO	RN RU
ISRLESS		2	000ED8	000136	00 FO	RN RU
ISRLMPP		2	0014D0	000B0A	00 FO	RN RU

Figure 61. Load Module Library Display - Expanded Line Command Field

Z - Compress Data Set

For the Z command, the processing is the same as if you had selected option C from the Library utility (3.1). This command recovers wasted space that was formerly occupied by deleted or updated members and is now available for use again.

The Z command invokes either the IEBCOPY utility or the compress request exit routine. See *Installation and Customization* for more information.

F - Free Unused Space

The F command releases space that is not being used by the data set. For example, if a data set is allocated with 100 tracks but is only using 60 tracks, the F command releases the 40 tracks that are not being used.

= - Repeat Last Command

The = command repeats the most recently used line command. This command is most helpful when the same TSO command or CLIST is to be invoked for more than one data set in a data set list. For example, suppose you have a CLIST named TESTABC and two data sets named USER.DATA1 and USER.DATA2. To run the CLIST with the two data sets consecutively from a data set list, you could:

1. Type "TESTABC" in the Command field beside USER.DATA1.
2. Type "=" in the Command field beside USER.DATA2.
3. Press the ENTER key.

This procedure saves keystrokes because you type the CLIST name only once and you press the ENTER key only once.

Data Set List Utility (Option 3.4)

Data Set List Utility Primary Commands

Six primary commands are available when you are using the Data Set List utility. These commands, which you type in the COMMAND field, are:

- CONFIRM
- FIND and RFIND
- LOCATE
- SAVE
- SHOWCMD
- SORT

The following sections describe these commands, which are shown in alphabetic order.

CONFIRM Command

The CONFIRM primary command controls display of the Confirm Delete panel. Use the following format:

```
CONFIRM  [ON|OFF]
```

You can use the following parameters with the CONFIRM command:

ON Tells PDF to display the Confirm Delete panel when you enter the D (delete data set) line command or TSO DELETE command. This is the default setting.

OFF Tells PDF not to display the Confirm Delete panel.

For example, the following command would tell PDF not to display the Confirm Delete panel:

```
CONFIRM OFF
```

FIND and RFIND Commands

The FIND primary command finds and displays a character string within the data set name. Use the following format:

```
FIND string [NEXT|PREV|FIRST|LAST|ALL]  
            [CHARS|WORD|SUFFIX|PREFIX]
```

For example, the following command would tell PDF to find all occurrences of the character string ELSE:

```
FIND ELSE
```

If you need more information about the parameters used with this command, see "FIND - Finding Character Strings" on page 90. NEXT and CHARS are the default parameters.

If necessary, PDF automatically scrolls to bring the character string to the top of the list. To repeat the search without re-entering the character string, use the RFIND command.

LOCATE Command

The LOCATE primary command scrolls the list of data sets based on the field on which the data set list is sorted, as described under “SORT Command” on page 146. Use the following format:

LOCATE *lparm*

You can use the following parameter with the LOCATE command:

lparm One of the following:

- If the list is sorted by data set name, specify a data set name.
- If the list is sorted by another field, specify a value for the field by which the list is sorted.

For example, for a data set list sorted by volume, you could enter the following:

This command locates the first data set in the list on volume TSOPK1. If the value is not found, the list is displayed starting with the entry before which the specified value would have occurred.

SAVE Command

The SAVE primary command writes the data set list to the ISPF list data set or to a sequential data set. PDF writes the data set list in its current sort order. Use the following format:

SAVE [*list-id*]

where:

list-id The name of a sequential data set. PDF names the data set:

prefix.userid.list-id.DATASETS

where:

prefix Your TSO user prefix if you have one and if it is different from your user ID.

userid Your TSO user ID.

The data set is created if it does not exist, or written over if it exists, and has compatible attributes. If you omit the *list-id* parameter, PDF writes the list to the ISPF list data set and includes the list headings.

The following command would tell PDF to write the list to a sequential data set named either *prefix.userid.MY.DATASETS* or *userid.MY.DATASETS*.

COMMAND ==> *SAVE*

Data Set List Utility (Option 3.4)

If the sequential data set already exists, PDF writes over it; if not, PDF creates it.

PDF displays a message, if necessary, when you enter a SAVE or SORT primary command that shows that the data needed to do the function is being retrieved. The keyboard locks when this message appears and stays locked until the SAVE or SORT command is completed.

SHOWCMD Command

The SHOWCMD primary command controls the display of line commands and their arguments as they are invoked. Use the following format:

```
SHOWCMD  [ON|OFF]
```

where:

ON Tells PDF to display line commands. This is the default setting.

OFF Tells PDF not to display line commands. Though the SHOWCMD default is ON, it is initially set to OFF.

After you enter SHOWCMD ON, a panel (Figure 62 on page 145) is displayed each time you enter a line command, TSO command, or CLIST on a data set list display.

On this panel, you see the command as you typed it and then, a few lines down, you see the command as PDF interpreted it. Seeing these commands displayed can be especially useful when you use the "/" character to substitute for the data set name because the panel shows the line command after expansion occurs. Therefore, you can tell immediately whether you need to add an argument(s) to the command.

For example, suppose you have a data set list displayed on the screen and decide to browse member MEMB1 of data set USER.TEST.DATA. To see how PDF will interpret the B (browse) line command, type SHOWCMD ON in the COMMAND field and press the ENTER key. Then, enter the following line command in the line command field to the left of USER.TEST.DATA:

```
B / MEMB1
```

When you press the ENTER key, the panel shown in Figure 62 on page 145 is displayed.

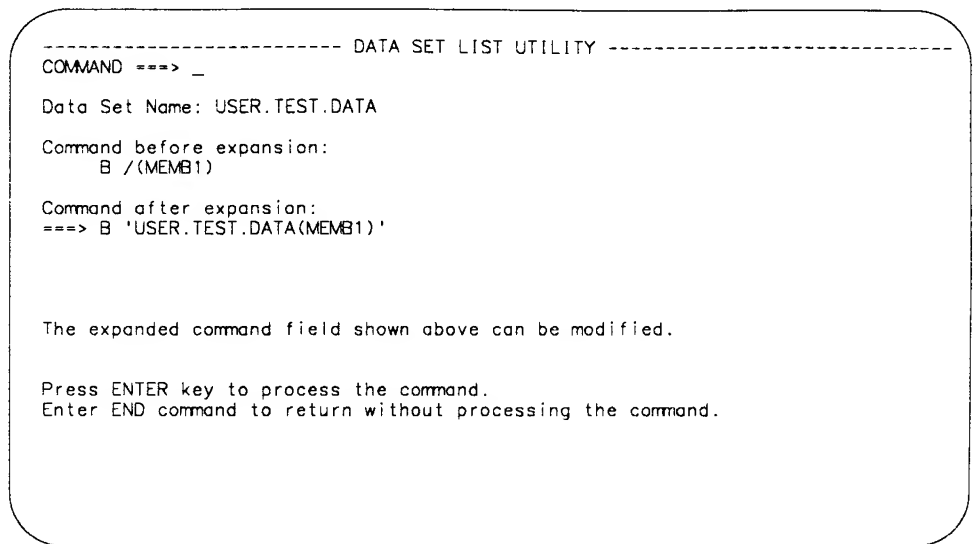


Figure 62. Data Set List Utility - SHOWCMD Panel

Notes:

- 1. The data set name and commands shown in Figure 62 are for example purposes only. These values are determined by the command you enter and the data set acted on by that command.*
- 2. SHOWCMD must be entered from a data set list. It is invalid if you use a line command, such as M to display a member list, before invoking it.*

When the panel showing the commands is displayed, you can do one of the following:

- Press the ENTER key to invoke the command displayed in the “Command after expansion” field.
- Change the command displayed in the “Command after expansion” field and then press the ENTER key to invoke the changed command.
- Enter the END command to return to the data set list display.

For information about using line commands, TSO commands, CLISTS, and the “/” character on a data set list display, see “Blank - Display Data Set List” on page 127.

Data Set List Utility (Option 3.4)

SORT Command

The SORT primary command sorts the data set list by the specified field. Use the following format:

```
SORT [field1 [field2]]
```

You can use the following parameters with the SORT command:

field1 The major sort field. If only one parameter is used, PDF treats it as field1. If both parameters are used, PDF sorts the list by field1 first, then by field2, within field1.

field2 The minor sort field.

For example, to sort a data set list by volume and block size within each volume, use the following command:

```
COMMAND ==>
```

If you do not specify a field, PDF sorts the list by data set name. The following keywords tell PDF by which field(s) to sort the data set list:

Field	Sequence	Description
NAME	Ascending	Data set name
MESSAGE	Ascending	Command completion message
VOLUME	Ascending	Volume serial
DEVICE	Ascending	Device type
DSORG	Ascending	Data set organization
RECFM	Ascending	Record format
LRECL	Descending	Logical record length
BLKSZ	Descending	Block size
TRACKS	Descending	Data set size
%USED	Descending	Percentage used
XT	Descending	Extents used
CREATED	Descending	Creation date
EXPIRES	Ascending	Expiration date
REFERRED	Descending	Last accessed data

Figure 63. SORT Fields for Source Libraries

Automatic scrolling will be performed, if necessary, to bring the major sort field into view. PDF displays a message when you enter a SAVE or SORT primary command that shows that the data needed to do the function is being retrieved. The keyboard locks when this message appears and stays locked until the SAVE or SORT command is completed.

Reset ISPF Statistics Utility (Option 3.5)

When you select this option, a panel is displayed to allow you to create, update, or delete statistics and to reset sequence numbers (Figure 64). The Reset ISPF Statistics utility handles only partitioned data sets with fixed- or variable-length records and logical record length of:

- From 10 to 255, inclusive, for fixed-length records
- From 14 to 259, inclusive, for variable-length records.

For information about ISPF statistics, see “ISPF Member Statistics” on page 39.

```
----- RESET ISPF STATISTICS -----
OPTION ==> R

R - Reset (create/update) ISPF statistics
D - Delete ISPF statistics

NEW USERID      ==>      (If userid is to be changed)
NEW VERSION NUMBER ==> 2  (If version number is to be changed)
RESET MOD LEVEL  ==> YES  (YES or NO)
RESET SEQ NUMBERS ==> YES  (YES or NO)

ISPF LIBRARY:
PROJECT ==> ISPFDEMO
GROUP   ==> XXX
TYPE    ==> COBOL
MEMBER  ==> * _          (Blank or pattern for member selection
                           list, '*' for all members)

OTHER PARTITIONED DATA SET:
DATA SET NAME ==>
VOLUME SERIAL ==>      (If not cataloged)
DATA SET PASSWORD ==>  (If password protected)
```

Figure 64. Reset ISPF Statistics Panel

All the fields on this panel are described in Chapter 2, “Libraries and Data Sets,” except the following:

NEW USERID

This field is used to set the USERID field in the statistics. Enter a new user ID here if you want to change the user ID the statistics will be recorded under. It is required if you do not specify a new version number.

If you are updating the user ID but not resetting the sequence numbers, the statistics are updated but the data is not scanned or renumbered.

NEW VERSION NUMBER

Enter a number here if you want to change the version number. This field is required if you do not specify a new user ID when resetting statistics. It is ignored if you are deleting statistics.

RESET MOD LEVEL

Enter YES if you want to reset the modification level or NO if you do not want to reset the modification level. A new version number is required to reset the modification level.

Reset ISPF Statistics Utility (Option 3.5)

RESET SEQ NUMBERS

Enter YES if you want to reset the sequence numbers or NO if you do not want to reset the sequence numbers. A new version number is required to reset the sequence numbers.

If the data is in packed format, there can be no sequence number processing. However, statistics for members in packed format can be created or updated if the sequence numbers are not being reset.

The following sections describe the options shown on the Reset ISPF Statistics panel.

R - Reset (Create/Update) ISPF Statistics

Use option R either to create statistics in a library that does not currently have them, or to update statistics in a library.

The NEW USERID field is optional for option R. If you specify a user ID, it is placed in the USERID field of the statistics. If you leave the NEW USERID field blank and select a member without statistics, the USERID field of the statistics is set to the current user ID.

Either a new user ID or a new version number is required when you use this option. When you specify a version number, the statistics are created or reset as follows:

Version Number	Set to the specified value.
Modification Level	Set to zero if requested; otherwise, unchanged.
Creation Date	Set to current date in the national format.
Change Date	Set to current date, in the national format, and time.
Current No. Lines	Set to the current number of data records.
Initial No. Lines	Set to the current number of data records.
No. Modified Lines	Set to zero if the value in the RESET SEQ NUMBERS field is YES.

If you have requested updating of the modification level and resetting of the sequence numbers, the last two digits of each sequence number are set to zeros. Otherwise, they are not changed.

If you have requested updating of sequence numbers, the data is scanned to determine if valid, ascending sequence numbers are present in all records. If so, the data is renumbered. Otherwise, the data is assumed to be unnumbered and renumbering is not done.

D - Delete ISPF Statistics

Use option D to delete ISPF statistics for an ISPF library or other partitioned data set. The NEW USERID and NEW VERSION NUMBER fields are ignored when you use option D.

Results of Resetting Statistics

What you specify for the NEW VERSION NUMBER, RESET MOD LEVEL, and RESET SEQ NUMBERS fields controls the resetting of the sequence numbers, the modification flags within the data, and the statistics. A new version number is required to reset the modification level and sequence numbers. Therefore, if a new version number is entered and the data is *not* in packed format, the following table shows the various combinations you can use for the RESET MOD LEVEL and RESET SEQ NUMBERS fields, and the results of those combinations:

RESET MOD LEVEL	YES	NO	YES	NO
RESET SEQ NUMBERS	YES	YES	NO	NO
Results:				
Modification flags	Reset	Unchanged	Unchanged	Unchanged
Sequence numbers	Reset	Reset	Unchanged	Unchanged

Figure 65. Reset Statistics Field Combinations and Results

Hardcopy Utility (Option 3.6)

Hardcopy Utility (Option 3.6)

The Hardcopy utility lets you specify a sequential data set or a member of a partitioned data set to be printed, and the destination of the output. It also lets you specify whether a sequential data set is to be kept or deleted after printing. Partitioned data set members are always kept.

You can use the Hardcopy utility to print any DASD-resident data set, except ISPF list and log data sets; use the ISPF LIST command to print log and list data sets during a PDF session.

An optional print utility exit can be specified by your installation. If this exit is installed, it may cause the Hardcopy utility's response to differ from the following descriptions. See *Installation and Customization* for more information about the print utility exit.

Another factor that can affect the Hardcopy utility's performance is whether the TSO/E Information Center Facility is installed. If the TSO/E Information Center Facility is installed, your installation can optionally allow PDF to display a panel for submitting TSO/E Information Center Facility information with the print request. See "Using the TSO/E Information Center Facility" on page 153 for more information.

If the TSO/E Information Center Facility is not installed, the Hardcopy utility first displays the panel shown in Figure 66.

```
----- HARDCOPY UTILITY -----
OPTION ==> PK

PK - Print and keep data set
PD - Print and delete data set

DATA SET NAME ==> 'COMPA.LIST'
VOLUME SERIAL ==> (If not cataloged)
DATA SET PASSWORD ==> (If password protected PDS)

PRINT MODE ==> BATCH (BATCH or LOCAL)

SYSOUT CLASS ==> A
LOCAL PRINTER ID ==>

JOB STATEMENT INFORMATION: (If not to local printer, verify before proceeding)
==> //useridA JOB (ACCOUNT),'NAME'
==> /*
==> /*
==> /*
```

Figure 66. Hardcopy Utility Panel - Before JCL Generation

Generating and Submitting JCL

Follow these steps to generate and submit JCL for your print job(s):

1. Choose one of the options listed at the top of the panel and type its code, PK or PD, in the OPTION field.
2. Enter a fully qualified data set name and member name.

You must specify at least the low-level qualifier, such as LIST. If you enter your user prefix as part of the data set name, you must enclose the complete character string in apostrophes. However, if you omit the user prefix and apostrophes, your user prefix is automatically added to the beginning of the data set name.

3. If the data set is not cataloged, enter the volume serial.
4. If your data set is password-protected, enter the password in the DATA SET PASSWORD field. See "Using Data Set Passwords" on page 49 if you need more information.
5. Enter either BATCH or LOCAL in the PRINT MODE field.

BATCH submits your print request as a background job.

LOCAL routes your data to a local printer, such as an IBM 328x printer that is connected to your terminal group.

6. Enter one of the following:
 - If you chose BATCH in the previous step, enter a SYSOUT CLASS and any job statement information you need.
 - If you chose LOCAL in the previous step, enter the name of a local printer in the LOCAL PRINTER ID field. Job statement information is ignored.

The SYSOUT CLASS, LOCAL PRINTER ID, and JOB STATEMENT INFORMATION fields are defined under Figure 9 on page 23.

7. Press the ENTER key.

What happens next depends on your choice in step 5. If you chose BATCH, see step 7a. If you chose LOCAL, see step 7b on page 152.

- a. If you chose BATCH, PDF generates the JCL and displays the panel shown in Figure 67 on page 152, with the message JCL GENERATED in the upper-right corner.

Hardcopy Utility (Option 3.6)

```
----- HARDCOPY UTILITY ----- JCL GENERATED
OPTION ==> PK

PK - Print and keep data set
PD - Print and delete data set
CANCEL - Exit without submitting job

Enter END command to submit job.

DATA SET NAME ==> 'COMPA.LIST'
VOLUME SERIAL ==> (if not cataloged)
DATA SET PASSWORD ==> (if password protected PDS)

PRINT MODE ==> (BATCH or LOCAL)

SYSOUT CLASS ==> A
LOCAL PRINTER ID ==>

JOB STATEMENT INFORMATION: (if not to local printer, verify before proceeding)
==> //useridA JOB (ACCOUNT),'NAME'
==> //*
==> //*
==> /*
```

Figure 67. Hardcopy Utility Panel - After JCL Generation

At this point you can either:

- Cancel the job by entering the CANCEL command in the OPTION field.
- Submit the job by entering the END command. PDF displays the following message at the bottom of the panel:

```
IKJ56250I JOB useridA(JOB nnnnn) SUBMITTED
***
```

Press the ENTER key. If you need more information about BATCH printing, see “Additional Batch Printing Information” on page 153.

- Enter another data set name for printing.
- b. If you chose LOCAL, ISPF invokes the DSPRINT TSO command processor to print the data set on the specified local printer.
- A message is displayed in the short message area to show that DSPRINT has accepted the request. At this point, you can:
- Specify another option and press the ENTER key.
 - Enter the END command.
 - Enter the CANCEL command.
8. If you entered CANCEL or END, PDF determines the next panel you see as follows:
- If you entered the Hardcopy utility from the ISPF/PDF Primary Option Menu or through the jump function (=), PDF displays the ISPF/PDF Primary Option Menu.

- If you entered the Hardcopy utility from the Utility Selection Menu, PDF returns you to that panel.

Additional Batch Printing Information

When you enter the desired information and press the ENTER key, PDF generates JCL that contains the job statement parameters and a job step that prints the specified data set, using the IBM IEBGENER utility.

Note: IEBGENER does not support packed data. If you try to print packed data, you may get unwanted results. IEBGENER prints the data set one logical record per print line. If the logical record length is greater than the printer width, the logical record is truncated.

PDF does not unpack data automatically before printing it. Therefore, if you need to unpack data before printing it, edit the data set and enter the PACK primary command with the OFF operand. See Edit and Edit Macros if you need more information about the PACK command.

Once the JCL for the first job step is generated, the job statement parameters are shown for information only. They are no longer intensified and you cannot type over them, since the job statement has already been generated. You can then select another data set name to cause another job step to be generated.

Using the TSO/E Information Center Facility

If the TSO/E Information Center Facility is installed, your installation can allow PDF to display the panel shown in Figure 68 on page 154.

Follow these steps to use the TSO/E Information Center Facility to submit your print job(s):

1. Choose one of the options listed at the top of the panel and type its code, PK or PD, in the OPTION field.
2. Enter a fully qualified data set name and member name.

You must specify at least the low-level qualifier, such as LIST. If you enter your user prefix as part of the data set name, you must enclose the complete data set name in apostrophes. However, if you omit the user prefix and apostrophes, your user prefix is automatically added to the beginning of the data set name.

3. If the data set is not cataloged, enter the volume serial.
4. If your data set is password-protected, enter the password in the DATA SET PASSWORD field. See "Using Data Set Passwords" on page 49 if you need more information.
5. You can either leave the PRINTER LOCATION field blank or enter the location ID of the printer to be used. The location ID is assigned by your installation.

Hardcopy Utility (Option 3.6)

```
----- HARDCOPY UTILITY -----
OPTION ===> PK

PK - Print and keep data set
PD - Print and delete data set

DATA SET NAME ===> 'COMPA.LIST'_
VOLUME SERIAL   ===>           (If not cataloged)
DATA SET PASSWORD ===>         (If password protected PDS)

PRINTER LOCATION   ===>
PRINTER FORMAT     ===>
NUMBER OF COPIES   ===>
```

Figure 68. Hardcopy Utility Panel - With the TSO/E Information Center Facility Installed

You can also enter a partial location name followed by an asterisk (*).

6. You can either leave the PRINTER FORMAT field blank or enter the format ID of the printer to be used. The format ID is assigned by your installation.

You can also enter a partial format name followed by an asterisk (*). If your printer location and format entries do not identify a specific printer, a printer selection list is displayed. From this list, which is similar to a member selection list, you can select a printer.

7. Specify the number of copies you want.
8. Press the ENTER key.

The values entered in the fields on this panel are passed directly to the TSO/E Information Center Facility for processing.

Removal of List VTOC Utility (Option 3.7)

The List VTOC utility (option 3.7) has been removed from Version 2 Release 3 of ISPF/PDF for MVS. To display or print VTOC (volume table of contents) information, use the V and PV options provided by the Data Set List utility (option 3.4).

Outlist Utility (Option 3.8)

Outlist Utility (Option 3.8)

This utility gives you the ability to browse, print, delete, or requeue job output that is in a held SYSOUT queue. When you select this option, a panel is displayed that allows you to select an option and enter the appropriate parameters (Figure 69).

```
----- OUTLIST UTILITY -----
OPTION ----> _

L - List job names/id's via the TSO STATUS command
D - Delete job output from SYSOUT hold queue
P - Print job output and delete from SYSOUT hold queue
R - Requeue job output to a new output class
blank - Display job output

FOR JOB TO BE SELECTED:
JOBNAME ----> KRAMERA
CLASS ---->
JOBID ---->

FOR JOB TO BE REQUEUED:
NEW OUTPUT CLASS ---->

FOR JOB TO BE PRINTED:
PRINTER CARRIAGE CONTROL ---->          (A for ANSI )
                                          (M for machine )
                                          (Blank for none)
```

Figure 69. Outlist Utility Panel

The fields on this panel are:

JOBNAME The held SYSOUT job. It is required for all options except option L.

CLASS The SYSOUT hold queue. If you omit the CLASS parameter, all SYSOUT queues are searched for the specified job.

JOBID Required only if more than one job exists with the same job name.

NEW OUTPUT CLASS

When requeuing a job (option R), enter the new SYSOUT hold queue here.

PRINTER CARRIAGE CONTROL

When printing a data set (option P), enter a value here that corresponds to the type of carriage control characters in the data set. Valid values are:

A If the data contains American National Standard Institute (ANSI) carriage control characters.

M If the data contains machine control characters.

Blank If the data contains no carriage control characters.

The record formats for the corresponding data sets are FBA, FBM, and FB, respectively.

The following sections explain the options listed at the top of the Outlist Utility panel.

L - List Job Names/IDs Via the TSO STATUS Command

If you select option L, a list of job names and job IDs is displayed. If you supply no job name, or if the job name is your ID plus one character, the status is displayed for all job names that are your ID plus one character. If you supply any other job name, the status for that exact job is displayed.

The list of job names is displayed on the lower portion of the panel. If the list is too long to fit on the screen, three asterisks are displayed on the last line of the screen. You can display the remainder of the list by pressing the ENTER key.

D - Delete Job Output from SYSOUT Hold Queue

If you select option D, the held output for a specific job is deleted from the specified SYSOUT queue.

P - Print Job Output and Delete from SYSOUT Hold Queue

If you select option P, the held output for a specific job is removed from the SYSOUT queue and placed in an ISPF-defined data set for printing. You can choose the record format for this data set by putting an entry in the PRINTER CARRIAGE CONTROL field.

An optional print utility exit can be installed by your system programmer. If this exit is installed, it may cause the Outlist utility's response to differ from the following descriptions. See *Installation and Customization* for more information about the print utility exit.

Another factor that can affect the Outlist utility's performance is whether the TSO/E Information Center Facility is installed. If the TSO/E Information Center Facility is installed, your installation can optionally allow PDF to display a panel for submitting the TSO/E Information Center Facility information with the print request. See Figure 98 on page 218 for an example of this panel and "Using the TSO/E Information Center Facility" on page 153 for information about the fields on this panel.

If the TSO/E Information Center Facility is not installed, the Outlist utility displays the panel shown in Figure 96 on page 210 when you press the ENTER key. Use this panel to tell PDF how and where the job output is to be printed.

Outlist Utility (Option 3.8)

R - Requeue Job Output to a New Output Class

If you select option R, the held output for a specific job is requeued to another SYSOUT class from the specified SYSOUT queue. You must enter the new SYSOUT class on the panel.

Blank - Display Job Output

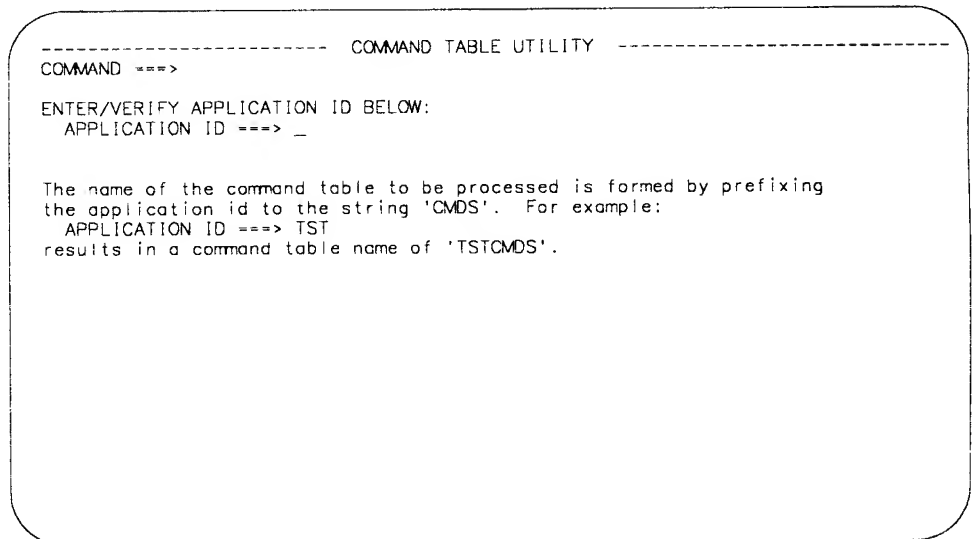
If you leave the OPTION field blank, the held output for the specified job is displayed in Browse mode. You can use all Browse commands. The data remains in the SYSOUT queue. When you enter the END or RETURN command to end Browse, the Outlist Utility panel is displayed again and you can then choose to print, requeue, or delete the job output.

Command Table Utility (Option 3.9)

The Command Table utility (3.9) allows you to create or change ISPF application command tables. When you select this option, a panel (Figure 70) is displayed to prompt you for an application ID. The name of the command table is then derived by adding CMDS to the application ID. If the table exists in the table input library, ISPTLIB, it is displayed and can be modified. If the table does not exist in the table input library, a new table is generated.

You cannot use this utility to change a command table that is currently in use. Command table ISPCMDS, the system command table, is always in use by the dialog manager. To change the system command table, make a copy of the table, rename the copy, change the copy, and then replace the original with the copy *outside* the ISPF environment. If you use the TSO COPY command, specify the NONUM parameter. Otherwise, you will have an unusable command table.

While you are using this utility to change a command table, the table cannot be used for other purposes. For example, you cannot use split screen and select a function with NEWAPPL(XYZ) if you are changing command table XYZCMDS.



```
----- COMMAND TABLE UTILITY -----
COMMAND --->

ENTER/VERIFY APPLICATION ID BELOW:
APPLICATION ID ---> _

The name of the command table to be processed is formed by prefixing
the application id to the string 'CMDS'. For example:
APPLICATION ID ---> TST
results in a command table name of 'TSTCMDS'.
```

Figure 70. Command Table Utility Panel

The only field on this panel is:

APPLICATION ID

The name of an application for which you want to define commands.

The command table for the named application is displayed on a Command Table editing panel (Figure 71 on page 160). This panel can be scrolled up and down using the scroll commands.

Command Table Utility (Option 3.9)

```
COMMAND TABLE - TSTCMDS ----- ROW 1 OF 17
COMMAND ----> _ SCROLL ----> PAGE

INSERT, DELETE, AND CHANGE COMMAND ENTRIES.  UNDERSCORES NEED NOT BE BLANKED.
ENTER END COMMAND TO SAVE CHANGES OR CANCEL TO END WITHOUT SAVING.

      VERB      T  ACTION
                DESCRIPTION

**** SORT      0  SELECT PGM(PQRSORT) PARM(&ZPARM)
                SORT ENTRIES BY ASCENDING ORDER
**** PREPARE   4  SELECT CMD(XPREP &ZPARM) NEWPOOL
                PREPARE FILE FOR FORMATTING
**** QUIT      2  ALIAS END
                QUIT COMMAND - SAME FUNCTION AS END
**** EXPLAIN   4  ALIAS HELP
                EXPLAIN COMMAND - SAME FUNCTION AS HELP
**** UP        0  &SCRVERT
                SCROLL UP COMMAND
**** DOWN      0  &SCRVERT
                SCROLL DOWN COMMAND
***** BOTTOM OF DATA *****
```

Figure 71. Command Table Editing Panel

The column headings on the panel are:

VERB	The command verb, which is the name of the command you are defining in the command table. A command verb must be from two to eight characters in length, inclusive, and must begin with an alphabetic character. The content of this column is assigned to the ZCTVERB system variable.
T (truncation)	The minimum number of characters that you must enter to find a match with the command verb. If this number is zero or equal to the length of the command verb, you must enter the complete command verb. For example, in Figure 71 the PREPARE command has a truncation value of 4. Therefore, for the TST application used as the example in the figure, only the first four letters, PREP, must be entered to invoke this command. The content of this column is assigned to the ZCTTRUNC system variable.
ACTION	The actual coding of the action to be carried out when you enter the command. The action length must not be greater than 60 characters. The content of this column is assigned to the ZCTACT system variable. The valid actions are: SELECT Causes the selected dialog (command, program, or selection panel) to be given control immediately. See <i>Dialog Management Guide</i> for more information about the SELECT statement and its keywords.

Command Table Utility (Option 3.9)

ALIAS	Allows one command verb to carry out the action defined for another. For example, in Figure 71 QUIT is an alias for END. Therefore, for the TST application used as the example in the figure, entering QUIT causes the same action to occur as entering END.
PASSTHRU	Causes the command to be passed through to the dialog, as if it had not been found in the command table.
SETVERB	Causes the command to be passed through to the dialog, with the command verb stored separately from the parameters.
NOP	Causes the command to be inoperative. An inactive command message is displayed.
Blank	Causes the command table entry to be ignored. PDF continues to search for additional entries for the same command verb. If the command is not found in either the application command table or the system command table, an invalid command message is displayed.
&xxxxxx	A variable name, beginning with an ampersand (&), allows dynamic specification of the command action.

Note: Do not use SPLIT, SPLITV, SWAP, CURSOR, PRINT, PRINTG, PRINTHI, PRINTL, or PRINTLHI as keywords in the ACTION column. These keywords are intended only for use in the system command table distributed with ISPF. They are not intended for use in application command tables.

DESCRIPTION	An optional, brief description of the action the command verb is to perform. Since this column is offset three spaces under the ACTION column, the description length must not be greater than 57 characters. The content of this column is assigned to the ZCTDESC system variable.
--------------------	--

For a new table, this panel initially contains dummy entries with all fields shown as underscores. The underscores are pad characters and need not be blanked out. However, any null entries where at least the verb contains all underscores are automatically deleted when the table is saved.

Command Table Utility (Option 3.9)

Scrolling a Command Table

You can scroll the table entries, using the ISPF UP and DOWN scroll commands, and change one or more entries simply by typing over them.

Saving a Command Table

The END command causes the table to be saved in the table output library, ISPTABL, and ends the utility.

Cancelling a Command Table

The CANCEL command ends the command table display without saving the table.

Command Table Line Commands

You can enter the following line commands at the left of any entry by typing over the four quotation marks. Multiple line commands or changes can be entered in a single interaction. Line commands followed by a number, such as D3, are repeated that number of times. The lines are processed in the order that they appear on the screen in. Any line commands or changes that are entered concurrently with the END command are processed before the table is saved.

D - Deleting Lines

The D command deletes one or n (Dn) lines.

I - Inserting Lines

The I command inserts one or n (In) lines. The inserted lines contain underscores (pad characters) in all field positions.

R - Repeating Lines

The R command repeats a line one or n (Rn) times. The repeated lines contain underscores (pad characters) in the VERB and T (truncation) fields, but the ACTION and DESCRIPTION fields are copied from the line on which the R or Rn was entered.

Convert Menu/Messages (Option 3.10)

This option is only for users who have installed the System Productivity Facility (SPF) licensed program, because old-format menus and messages can no longer be displayed with this release.

Installations that have previously extended or custom-tailored SPF must make sure that the ISPF/PDF Primary Option Menu and all lower-level selection panels (menus) that were displayed by the SPFUTIL program are in the new format. In ISPF, these menus are displayed by the SELECT service. The SPFUTIL program no longer exists.

This option automatically converts old-format SPF menus and messages to new-format ISPF panel and message definitions. When you select this option, two panels are displayed that are similar to the panels for the Move/Copy utility (3.3). The first panel (Figure 72) allows you to specify whether you are converting menus or messages, option 1 or 2, respectively, and to specify the “old format” data set, which must be partitioned.

```
----- CONVERT MENU/MESSAGES -----
OPTION  ===> 1

      1 - Convert old format menus to new format panels
      2 - Convert old format messages to new format messages

SPECIFY "OLD FORMAT" DATA SET BELOW.
THEN PRESS ENTER KEY TO SPECIFY "NEW FORMAT" DATA SET.

FROM ISPF LIBRARY:
PROJECT  ===> SPF22
GROUP   ===> OURMODS
TYPE    ===> MENUS_
MEMBER   ===>                                     (Blank for member selection list,
                                                         '*' for all members)

FROM OTHER PARTITIONED DATA SET:
DATA SET NAME  ===>
VOLUME SERIAL  ===>                                     (If not cataloged)

DATA SET PASSWORD ===>                                     (If password protected)
```

Figure 72. Convert Menu/Messages Panel

The fields on this panel are described in Chapter 2, “Libraries and Data Sets” on page 29.

The second panel, shown in Figure 73 on page 164, allows you to specify the “new format” data set, which also must be partitioned.

Note: Old-format members are limited to a maximum of 180 input records.

Members are read from the “old format” data set, converted to the new format, and stored in the “new format” data set. This utility does not change the “old format” data set.

Members in the “old format” data set can be in packed or unpacked format, but data in the “new format” data set is *always* in unpacked format.

Convert Menus/Messages Utility (Option 3.10)

```
COPY --- OLD FORMAT SPF22.OURMODS.MENUS -----
COMMAND ---->

SPECIFY "NEW FORMAT" DATA SET BELOW.

TO SPF LIBRARY:
PROJECT ----> ISP
GROUP ----> OURMODS
TYPE ----> ISPLIB_

TO OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ---->
VOLUME SERIAL ----> (If not cataloged)

DATA SET PASSWORD ----> (If password protected)

REPLACE LIKE-NAMED MEMBERS ----> (YES or NO)
```

Figure 73. Copy Panel for Specifying New Format Data Set

The following sections describe the options shown at the top of the Convert Menus/Messages panel.

1 - Convert Old Format Menus to New Format Panels

This utility is not capable of fully converting all old-format SPF menus to ISPF panels. In general, this utility:

- Handles conversion of attribute characters, including those specified with the old-format <FIELDS> statement, if the <FIELDS> statement is the first line found in the old-format menu
- Generates an)ATTR section, if required, and a)BODY section. The name of the first input field on line 2, if any, is changed to ZCMD. Other input and output fields retain the same names as in the old-format menu.
- Generates an)INIT section with .HELP set to the name specified using the old GENHELP keyword, and with variables initialized as specified using the old INIT keywords.

The Convert Menus/Messages utility (3.10) does *not* handle explicit cursor placement, the old KEY keyword, or the old LIST/RETURN keywords except as described below.

Convert Menus/Messages Utility (Option 3.10)

The Convert Menus/Messages utility (3.10) *fully* converts:

- Old-format selection menus *below the primary option level* that were designed to be processed by the SPFUTIL program in the System Productivity Facility
- Old-format tutorial pages, except that the utility cannot handle the bypass of a tutorial page that is displayed only if explicitly selected. The converted tutorial page would *not* be bypassed if you were to run the tutorial and kept pressing the ENTER key. To correct this problem, you must manually change the new-format parent panel by inserting an asterisk before the panel name in the TRANS statement. For more information, see *Dialog Management Guide*.

2 - Convert Old Format Messages to New Format Messages

For messages, the utility properly handles the text of both short and long messages, the HELP reference, if any, and the ALARM indicator. Variable fields in old-format messages are converted to dummy variable names, which begin with an ampersand. You must change these manually to the appropriate dialog variable names.

Format Specification Utility (Option 3.11)

Format Specification Utility (Option 3.11)

The Format Specification utility (3.11) is provided to support the IBM 5550 terminal using the Double-Byte Character Set (DBCS). It is used to maintain formats that are used when editing and browsing to display data sets that contain predefined formatted records.

The purpose of a format is to structure data from a record into fields and to define the order these fields are to be physically displayed in on the screen when you are editing and browsing.

When you select this option, a panel (Figure 74) is displayed that allows you to add, copy, delete, or update a format. You can also display the format list.

```
----- FORMAT SPECIFICATION -----
OPTION ----> _

A          - Add a new format
C          - Copy formats
D          - Delete a format
U          - Update a format
L or BLANK - Display format list

FORMAT NAME ---->

FOR COPY OPERATIONS, SPECIFY THE FOLLOWING:

FROM FORMAT ---->          (Blank for format list, * for all formats)
FROM TABLE ---->          (Default is "ISRFORM")

NOTE: The Format Utility is provided for support of the IBM 5550 terminal
      using the Double Byte Character Set (DBCS).
```

Figure 74. Format Specification Panel

The fields on this panel are:

FORMAT NAME

The name of the format that you want to add, delete, or update. When copying a format (option C), this is the name you want the copied format stored under.

FROM FORMAT

When copying a format (option C), you can:

- Enter the name of a format you want to copy.
- Enter an asterisk (*) to copy all formats.
- Leave the field blank to display a copy format selection list.

See Figure 76 on page 169 for an example of a Copy Format Selection List display.

Format Specification Utility (Option 3.11)

FROM TABLE

When copying a format (option C), you can:

- Enter the name of a table from which you want to copy a format.
- Leave the field blank if you want to copy a format from the ISRFORM table.

The following sections describe the options shown at the top of the Format Specification panel.

A - Add a New Format

If you specify option A and a format name, the Format Definition panel (Figure 75) is displayed.

----- FORMAT DEFINITION (FORM01) -----

COMMAND ===> _

FIELD NUMBER	START COLUMN	FIELD LENGTH	FIELD TYPE	FIELD NUMBER	START COLUMN	FIELD LENGTH	FIELD TYPE
1	000	00	—	2	000	00	—
3	000	00	—	4	000	00	—
5	000	00	—	6	000	00	—
7	000	00	—	8	000	00	—
9	000	00	—	10	000	00	—
11	000	00	—	12	000	00	—
13	000	00	—	14	000	00	—
15	000	00	—	16	000	00	—
17	000	00	—	18	000	00	—
19	000	00	—	20	000	00	—

FIELD NUMBER: Identifies the field position on the screen.
START COLUMN: Specifies the column position in the record.
FIELD LENGTH: From 1 to 71; fields must not overlap.
FIELD TYPE : E - single-byte, D - double-byte, M - mixed data

Enter the END command to exit and save the format.
Enter the CANCEL command to exit without saving the format.

Figure 75. Format Definition Panel

A field definition includes:

FIELD NUMBER

The number of the field for which you are defining a format. You can define up to 20 fields.

START COLUMN

Starting column position in the record.

FIELD LENGTH

Field length in bytes; the maximum is 71 bytes.

FIELD TYPE

The type of data that can be entered in the field. Valid values are:

E	EBCDIC (single-byte)
D	DBCS (double-byte)
M	Mixed data.

Format Specification Utility (Option 3.11)

Note: All three of these field types can contain extended graphics characters. CAPS ON processing is not possible because of context dependencies. Therefore, it is ignored when you are editing formatted data.

The format definition information applies to both existing records and inserted records in a data set.

Note: We recommend that you avoid using formats with numbered data, either STD or COBOL. The results can be different from using formats with unnumbered data. If you must use numbered data, do not define the columns the sequence numbers will appear in, or define an EBCDIC or mixed data field for them.

C - Copy Formats

If you specify option C in the Format Specification panel, the following occurs:

- If you specify both an asterisk (*) in the FROM FORMAT field, and a table name other than ISRFORM in the FROM TABLE field, all formats cataloged in the “from” table are copied to ISRFORM.

Note: If you specify a table name in the FROM TABLE field, and that table does not have the same format as ISRFORM, a severe error occurs.

- If you specify both a format name and a “from” format, the format is copied. If you specified a “from” table (other than ISRFORM), the format is copied from that table. Otherwise, the format is copied from ISRFORM. The Format Definition panel for the newly created format, containing the currently defined fields, is displayed. You can add, delete, and/or update field definitions. When you press the END key, the format definition is cataloged in ISRFORM under the format name you specified.
- If you specify a format name but no “from” format, the Copy Format Selection List panel (Figure 76 on page 169) is displayed.

If you did not specify a “from” table, the formats listed are those cataloged in ISRFORM, the default format table. Otherwise, the formats listed are those cataloged in the table you specified.

You can select a format to copy by entering S to the left of that format name. Other commands you can enter are U (Update), R (Rename), D (Delete), SELECT (which is similar to S), SORT, and LOCATE. See “Format Selection List Commands” on page 170 for a description of these commands.

The format is copied, and the Format Definition panel for the newly created format, containing the currently defined fields, is displayed. You can add, delete, and/or update field definitions. When you press the END key, the format definition is cataloged in ISRFORM under the format name you specified on the Format Specification panel.

----- COPY FORMAT SELECTION LIST (ISRFORM) -----						ROW 1 OF 4
COMMAND ==>	NAME	RENAME	CREATED	LAST MODIFIED	ID	SCROLL ==> PAGE
S	COMMON1		85/05/02	85/05/06 10:13	Z72KXX	
	COMMON2		85/05/02	85/05/06 10:14	Z72KXX	
	COMMON3		85/05/02	85/05/06 10:14	Z72KXX	
S_	COMMON4		85/05/02	85/05/06 10:16	Z72KXX	
***** BOTTOM OF DATA *****						

Figure 76. Copy Format Selection List Panel

- If you specify neither a format name nor a “from” format, but you do specify a “from” table (other than ISRFORM), the Copy Format Selection List panel is displayed. You can select one or more formats to copy by entering **S** to the left of the format name(s). Each of these formats is copied under the same name from the specified “from” table to the ISRFORM table.

Note: If you do not specify option C but specify a name in the FROM FORMAT field, the FROM FORMAT field is ignored.

D - Delete a Format

If you specify option D and a format name in the Format Specification panel, the format is deleted.

U - Update a Format

If you specify option U and a format name in the Format Specification panel, the Format Definition panel containing the currently defined fields is displayed. You can add, delete, and/or update field definitions.

L or BLANK - Display Format List

If you specify option L or leave the OPTION line blank in the Format Specification panel, the Format Selection List panel (Figure 77 on page 170) is displayed.

Format Specification Utility (Option 3.11)

----- FORMAT SELECTION LIST -----						ROW 1 OF 4
COMMAND --->						SCROLL ---> PAGE
U_	NAME	RENAME	CREATED	LAST MODIFIED	ID	
	FORM01		85/05/02	85/05/06 10:13	Z72KXX	
	FORM02		85/05/02	85/05/06 10:14	Z72KXX	
	FORM03		85/05/02	85/05/06 10:14	Z72KXX	
	FORM04		85/05/02	85/05/06 10:16	Z72KXX	
***** BOTTOM OF DATA *****						

Figure 77. Format Selection List Panel

Format Selection List Commands

The following sections describe the commands you can use on a Format Selection List panel.

Deleting a Format

If you specify the D line command beside a format name, the format is deleted.

Locating Format Names

The LOCATE command is another useful tool, especially if you have a long format list. To use the LOCATE command, ensure that the list is sorted by name. Next, enter the word **LOCATE** followed by the format name on the COMMAND line. The format is:

LOCATE name

where:

name The name of the format you want to find.

For example, the following command would find a format named FORM03:

LOCATE FORM03

If the format exists, the entry for the specified format name appears as the second line following the header lines. If the specified name is not found, the existing format name that immediately precedes the specified name appears as the first line following the header lines.

Renaming a Format

If you specify the R line command beside a format name, you must also specify its new name in the RENAME field before you press the ENTER key. If you do not, the ENTER REQUIRED FIELD message appears in the upper-right corner of the screen and the cursor moves to the RENAME field.

Sorting Format Names

You can sort the name-list on this panel by entering the SORT command on the COMMAND line. The format of the SORT command is:

```
SORT [NAME|TIME]
```

where:

NAME	Sort by name.
TIME	Sort by time last modified.

For example, the following command would sort a format selection list by time:

Updating or Selecting a Format

If you specify the U or S line command beside a format name, the Format Definition panel containing the currently defined fields is displayed. You can add, delete, and/or update field definitions.

You can specify that multiple operations be done at the same time. However, if you specify U or S with other line commands, any commands after the first U or S are ignored.

The SELECT command provides you with another way to specify a format. This command is entered on the COMMAND line.

The format of the SELECT command is:

```
SELECT name
```

where:

name	The name of the format you want to select.
-------------	--

If the format exists, the Format Definition panel containing the currently defined fields is displayed. You can add, delete, and/or update field definitions.

If no format exists for that name, a new format is created, and the Format Definition panel is displayed to allow you to define fields.

SuperC Utility (Option 3.12)

SuperC Utility (Option 3.12)

Note: For an introduction to the SuperC and SuperCE utilities (options 3.12 and 3.13), see Appendix E, "SuperC Reference."

The SuperC utility (3.12) is a dialog that uses the SuperC program to compare data sets of unlimited size and record length at the file, line, word, or byte level. The panel shown in Figure 78 is used to specify the name of a "new" data set.

Note: In this context, a "new" data set is an updated version of a previously created data set, such as a data set in your private library that has not yet been promoted.

```
----- SUPERC UTILITY -----
COMMAND ==>

SPECIFY "NEW" DATA SET TO BE COMPARED, THEN PRESS THE ENTER KEY.

PROJECT ==> FBROWN
GROUP   ==> PRIVATE   ==>           ==>
TYPE    ==> TABLES
MEMBER  ==> _          (Blank or pattern for member selection list,
                        '*' for all members)

"NEW" OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ==>
VOLUME SERIAL ==>      (If not cataloged)

DATA SET PASSWORD ==>      (If password protected)

PROFILE DSN      ==>

MODE             ==> F      (F - foreground, B - batch)
```

Figure 78. SuperC Utility Panel

This panel requires only the names of the input data sets, and a foreground or batch mode setting.

All the fields on this panel are explained in Chapter 2, "Libraries and Data Sets," except:

PROFILE DSN

The name of an optional data set that can contain a compare type, listing type, sequence numbers setting, Browse setting, process options, and process statements. All these elements, when combined in one data set or member, are called a *profile*. See "A - Activate/Create Profiles" on page 195 for information about using the SuperCE utility (3.13) to create a profile data set.

The listing type and sequence numbers setting of the profile are copied onto the panel used to specify the "old" data set name (Figure 79 on page 173), but can be typed over or blanked out. However, other elements of the profile are in effect, even though they are not shown on the panel.

MODE The processing mode you want to use when comparing the data sets. Choose one of the following:

- F** Foreground mode. After the “old” data set panel and member selection, if any, are completed, foreground mode compares the “new” and “old” data sets and stores the results in the data set specified in the LISTING DS NAME field, which you can browse at the terminal.
- B** Batch mode. After the “old” data set panel and the member list, if any, are completed, batch mode causes the display of the SuperC Utility - Submit Batch Jobs panel, so you can specify job card and print disposition information. Then, JCL is generated and a batch job is submitted to compare the “new” and “old” data sets.

Note: You cannot specify a data set password in batch mode. If your data sets are password-protected, use foreground mode.

When you complete the “new” data set panel and press the ENTER key, PDF displays the panel shown in Figure 79. Of the five fields shown at the bottom of the panel, all except LISTING TYPE may not appear, depending on the mode you choose (foreground or batch) and the contents of the profile data set. Also, if you request a member list or specify an asterisk (*) in the MEMBER field on the “new” data set panel, PDF does not display a MEMBER field on the “old” data set panel.

```

COMPARE -- FBROWN.PRIVATE.TABLES -----
COMMAND ---->

SPECIFY "OLD" DATA SET TO BE COMPARED. THEN PRESS THE ENTER KEY.

PROJECT ----> FBROWN
GROUP  ----> PRIVATE ---->          ---->
TYPE   ----> TABLES
MEMBER ----> _

"OLD" OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ---->
VOLUME SERIAL ----> (If not cataloged)

DATA SET PASSWORD ----> (If password protected)
UPDATE DS NAME ---->
LISTING TYPE ----> DELTA (DELTA/CHNG/LONG/OVSUM/NOLIST)
LISTING DS NAME ----> SUPERC.LIST
SEQUENCE NUMBERS ----> (blank/SEQ/NOSEQ/COBOL)
  
```

Figure 79. SuperC Utility - “Old” Data Set Panel

SuperC Utility (Option 3.12)

Specify the name of an “old” data set. The type of “old” data set that you can specify depends on the type of “new” data set you specified on the previous panel. For example, you can compare:

- A complete “new” PDS to a complete “old” PDS
- A “new” sequential data set to:
 - An “old” sequential data set
 - An “old” membered PDS.
- A “new” membered PDS to an “old” sequential data set

In this context, the term *membered PDS* refers to a PDS for which a single member has been specified, such as:

SuperC treats a membered PDS as a sequential data set because the comparison is done on a one-to-one basis. However, SuperC cannot compare a sequential data set to a complete PDS because it cannot compare one data set to more than one member of another data set.

When you press the ENTER key, PDF either displays a member selection list or begins the comparison. All the fields on this panel are explained in Chapter 2, “Libraries and Data Sets,” except the following:

UPDATE DS NAME

Tells SuperC the name of the data set that will contain column-oriented results of the comparison. This data set is normally used as input to post processing programs and can be specified besides the normal listing data set. Also, this field is not displayed unless your profile data set contains an update (UPDxxxx) process option. See “P - Select SuperCE Process Options” on page 194 for information about the SuperC process options.

If you leave this field blank, SuperC uses the following default name:

`prefix.userid.SUPERC.UPDATE`

where “prefix” is your TSO prefix and “userid” is your user ID. If your prefix and user ID are identical, only your prefix is used. Also, if you do not have a prefix, only your user ID is used.

If you enter the name of a data set that already exists, the contents of that data set are replaced by the new update output.

If you enter the name of a data set that does not exist, SuperC allocates it for you. The data set is allocated as a sequential data set unless you enter a member name after it, in which case it is allocated as a partitioned data set.

NW: 15457 DocId: 70002077 Page 561 *Note: For the UPDMVS8, UPDCMS8, UPDSEQ0 and UPDPDEL process options, the update data set contains*

valid data, but only after a successful compare when differences are detected. The data set is always empty after a comparison that shows the data sets or members being compared have no differences.

LISTING TYPE

The type of listing you want SuperC to create when it compares the data sets. This is a required field, so you must choose one of the following. See "SuperC Listings" on page 344 for sample listings.

- | | |
|---------------|--|
| DELTA | Lists the differences between the source data sets, followed by the overall summary. |
| CHNG | Lists the differences between the source data sets, plus up to 10 matching lines before and after the differences. This listing is a variation of the DELTA listing; the matching lines before and after help you recognize changed areas of the source data sets. |
| LONG | Lists all the "new" data set source lines, plus "old" data set deleted lines. Both inserted and deleted lines are flagged. |
| OVSUM | Lists only the overall summary of the comparison. However, a PDS comparison generates an individual summary line for each PDS member. |
| NOLIST | Produces no listing output. In foreground mode, only a message is returned to show the outcome of the compare. |

LISTING DS NAME

The name of the list data set to which SuperC writes the results of the comparison. However, if you enter NOLIST in the LISTING TYPE field, SuperC does not create an output listing, so this name is ignored. Also, if you chose batch mode, this field does not appear on the panel. The SuperC Utility - Submit Batch Jobs panel is used instead.

If you leave this field blank, SuperC allocates a list data set, using default data set attributes and the following data set name:

`prefix.userid.SUPERC.LIST`

where "prefix" is your TSO prefix and "userid" is your user ID. If your prefix and user ID are identical, only your prefix is used. Also, if you do not have a prefix, only your user ID is used.

If you enter the name of a data set that already exists, the contents of that data set are replaced by the new output listing. However, if the data set is sequential, you can add this listing to the data set instead of replacing it by including the APNDLST process option in your profile data set.

SuperC Utility (Option 3.12)

If you enter the name of a data set that does not exist, SuperC allocates it for you. The data set is allocated as a sequential data set unless you enter a member name after it, in which case it is allocated as a partitioned data set.

SEQUENCE NUMBERS

A value that tells SuperC whether to exclude sequence number fields from its comparison of your data sets. This field is not displayed if the compare type is FILE or BYTE. The following choices tell SuperC to:

blank	Exclude sequence number fields from the comparison if the data set is FB 80 or VB 255. Otherwise, treat as data.
SEQ	Exclude sequence number fields from the comparison. Sequence numbers are assumed in columns 73-80 in FB 80 and in columns 1-8 in VB 255 data sets.
NOSEQ	Treat FB 80/VB 255 standard sequence number columns as data.
COBOL	Ignore columns 1-6 in FB 80 data sets. Data in columns 1-6 is assumed to be sequence numbers.

SuperC Member Lists

The panel shown in Figure 80 on page 177 is displayed after you specify the “old” data set name, but only if all the following statements are true:

- The “new” data set(s) is partitioned.
- The MEMBER field, shown on the SuperC Utility panel (Figure 78 on page 172), or the PDS Member List field, shown on the SuperCE Utility panel (Figure 83 on page 182) was left blank or a pattern was used. See “Displaying Member Lists” on page 38 for more information about displaying member selection lists.
- The profile data set or statements data set being used does not contain any SELECT process statements.

The members displayed in this list are members in the “new” data set. If the OLD-MEM column is blank, SuperC assumes each member in the “new” data set is to be compared with a member of the same name in the “old” data set.

If you enter a member name in the OLD-MEM column, SuperC compares this member to the one listed beside it in the NEW-MEM column.

To compare your selections, enter the END command. If you have not selected any members, PDF returns you to the previous panel.

```

COMPARE -- FBROWN.PRIVATE.TABLES ----- ROW 00001 OF 00008
COMMAND ----> _                               SCROLL ----> PAGE

Enter END command to process selections or CANCEL to exit the member list.
Enter OLD MEMBER name if it is different from NEW MEMBER.

NEW-MEM  OLD-MEM  VV.MM  CREATED  CHANGED  SIZE  INIT  MOD  ID
ISRUC1A  01.28  86/01/22  86/03/10  14:15   24   24   0  FBROWN
ISRUC1B  01.27  86/01/22  86/03/10  14:40   24   21   3  FBROWN
ISRUC1C  01.10  86/01/30  86/03/10  14:40   24   25   4  FBROWN
ISRUC1S  01.11  86/02/19  86/03/10  14:51   23   17   2  FBROWN
ISRUC1T  01.24  86/01/30  86/03/10  16:21   25   15   5  FBROWN
ISRUC2   01.28  86/02/13  86/03/10  14:30   24   20   2  FBROWN
ISRUC6L  01.10  86/02/25  86/03/10  12:19   24   21   1  FBROWN
ISRUC7   22.09  84/08/07  86/03/10  12:06   21   39   3  FBROWN
**END**

```

Figure 80. SuperC Member List Panel

To cancel your selections, enter either:

- The RESET command to remove all unprocessed selections without ending the member list display
- The CANCEL command to end the member list display without processing selections that are still on the screen.

Note: Both the jump function (=) and the RETURN command cause an implied cancellation of selections before they are implemented.

If you need more information about member lists, see “Using Member Selection Lists” on page 37.

Submitting a SuperC Job in Batch Mode

If you selected batch mode on the SuperC Utility panel, the panel shown in Figure 81 on page 178 is displayed before the job is submitted. This panel lets you specify one of the following:

- The SYSOUT CLASS, which determines the printer to which your job is sent and the format used for the printed output
- The name of a listing data set
- Output data definitions that you can use to give the printer additional instructions, such as an output destination that is not defined by a SYSOUT class.

SuperC Utility (Option 3.12)

```
----- SUPERC UTILITY - SUBMIT BATCH JOBS -----
OPTION  ----> _

blank  - Generate output listing to SYSOUT CLASS below.
  1    - Generate output listing to DATA SET NAME below.
  2    - Generate output listing using completed //OUTDD DD  below.

SYSOUT CLASS ----> A

DATA SET NAME ----> FBROWN.PRIVATE.TABLES

//OUTDD DD  ----> SYSOUT=H,FCB=12,CHARS=GT15
//          ---->
                                     LRECL for the Listing Output will be 133

JOB STATEMENT INFORMATION: (Required - Enter/Verify JOB control statements)
----> //FBROWNA JOB (ACCOUNT),'NAME'
----> /*
----> /*
----> /*
```

Figure 81. SuperC Utility - Submit Batch Jobs Panel

The JOB STATEMENT INFORMATION field is explained under Figure 9 on page 23. The other fields on this panel are:

Note: The following three fields are independent of one another. Also, none of them require you to provide an OUTDD card in the JOB STATEMENT INFORMATION field.

SYSOUT CLASS

A system output classification defined by your installation, which defines certain print characteristics, such as the printer and the format that will be used to produce the output. This field is required if you leave the OPTION field blank. Otherwise, it is ignored. See “Blank - Generate Output Listing to SYSOUT CLASS” on page 179 if you need more information.

DATA SET NAME

The name of the listing data set that you want PDF to store your compare results in. This data set can be either partitioned or sequential.

The DATA SET NAME field is required if you use option 1 on this panel. Otherwise, it is ignored. See “1 - Generate Output Listing in DATA SET NAME” on page 179 if you need more information.

//OUTDD DD

Output data definitions that are used to specify additional printer instructions in job control language (JCL). This field is required if you use option 2 on the SuperC Utility - Submit Batch Jobs panel. Otherwise, it is ignored. See “2 - Generate Output Listing Using Completed //OUTDD DD” on page 180 if you need more information.

The following sections describe the options shown at the top of the SuperC Utility - Submit Batch Jobs panel (Figure 81).

Blank - Generate Output Listing to SYSOUT CLASS

Leave the OPTION field blank if you want to send the listing output to a SYSOUT class. Specify the desired SYSOUT class in the SYSOUT CLASS field.

If you enter either option 1 or option 2, the SYSOUT CLASS field is ignored. However, for option 2, you can include the "SYSOUT =" parameter in an OUTDD DD field.

1 - Generate Output Listing in DATA SET NAME

Enter option 1 if you want SuperC to store the listing output in a data set. Specify the desired data set in the DATA SET NAME field. This field serves the same purpose as the LISTING DS NAME field, which is used when running the SuperC utility in foreground mode.

The logical record length (LRECL) of the listing data set is displayed under the list of options at the top of the SuperC Utility - Submit Batch Jobs panel. SuperC creates listings with either of four LRECLs:

- | | |
|-----|--|
| 132 | Standard listing for the NOPRTCC process option; printer control characters are omitted. |
| 133 | Standard listing. |
| 202 | Wide listing for the NOPRTCC process option; printer control characters are omitted. |
| 203 | Wide listing. |

If you specify an existing sequential data set with an incorrect LRECL, SuperC overrides the data set specifications. This applies to any listing and update data sets in both foreground and batch.

A separate operation, such as using the Hardcopy utility (option 3.6), is needed to print the listing data set.

If you leave the OPTION field blank or enter option 2, the DATA SET NAME field is ignored. Therefore, to specify an output data set in either of these two situations, you must include the "DSN =" parameter in an OUTDD DD field.

Specifying Existing Data Sets:

The following rules apply when specifying the name of an existing data set:

- When you submit JCL for processing, the output listing produced by that JCL usually replaces the contents of the specified data set, if any exist. Therefore, be careful when specifying the name of an existing data set.

SuperC Utility (Option 3.12)

You can keep a history of changes by using the APNDLST compare option when you run the comparison. This compare option adds the new output listing to the specified sequential data set's contents instead of replacing it.

- Use standard TSO data set naming conventions.

Specifying Nonexistent Data Sets:

The following rules apply when specifying the name of a data set that does not exist:

- If you include a member name in the data set specification, PDF allocates a partitioned data set with suitable attributes for the listing.
- If you do not specify a member name, PDF allocates a sequential data set.

2 - Generate Output Listing Using Completed //OUTDD DD

Enter option 2 if you want to specify an output data definition statement in the //OUTDD DD fields.

The OUTDD DD fields are provided so you can pass to your printer all the JCL needed to format special types of output that may not be supported by your installation's SYSOUT CLASS definitions. The example shown in Figure 81 on page 178 specifies a wide format for printing on 14 3/4 inch forms.

The SYSOUT CLASS and DATA SET NAME fields are ignored. If you need to specify this information, be sure to include it in your OUTDD DD job card. If you specify a data set name in your OUTDD DD job card, the output data set is printed and kept. Otherwise, it is printed and deleted. Here are some examples:

- To specify a SYSOUT class, enter:

```
//OUTDD DD SYSOUT=X
```

where "X" is the SYSOUT class, such A, B, or C.

- To specify a data set name, enter:

```
//OUTDD DD DSN=fully.qualified.name  
DISP=XXXXX...
```

where "XXXXX..." is one of the following:

— For an "old" data set:

— For a "new" sequential data set:

- For a "new" partitioned data set
- For a sequential data set that will be modified by, instead of replaced by, the comparison results:

Using the NOLIST Listing Type in Batch Mode

If you enter the NOLIST listing type and then choose batch mode, the options on the SuperC Utility - Submit Batch Jobs panel shown in Figure 81 on page 178 are invalid because no listing is produced. Therefore, an alternate panel is displayed, which blanks out the invalid fields, but still allows you to submit job statement JCL. This panel is shown in Figure 82.

```

----- SUPERC UTILITY - SUBMIT BATCH JOBS -----
OPTION ====> _

                "NOLIST" listing option selected.
                ENTER to SUBMIT job.   END to CANCEL.


JOB STATEMENT INFORMATION: (Required - Enter/Verify JOB control statements)
----> //FBROWNA  JOB (ACCOUNT),'NAME'
----> //*
----> //*
----> /*
  
```

Figure 82. SuperC Utility - Submit Batch Jobs Panel using NOLIST

When this panel is displayed, you can either:

- Type the job statement JCL and press the ENTER key to submit the job.
- Enter the END command to cancel the job.

SuperCE Utility (Option 3.13)

SuperCE Utility (Option 3.13)

Note: For an introduction to the SuperC and SuperCE utilities (options 3.12 and 3.13), see Appendix E, "SuperC Reference" on page 367.

The SuperCE utility (3.13) is a dialog that uses the SuperC program to compare data sets of unlimited size and record length at the file, line, word, or byte level. It is appropriate if you need more flexibility than the standard SuperC utility (3.12) provides.

The panel shown in Figure 83 is the first panel of the SuperCE utility. It requires only the names of the input data sets, which are entered using standard TSO naming conventions, such as:

New DS Name ===>

Note: In this context, a "new" data set is an updated version of a previously created data set, such as a data set in your private library that has not yet been promoted.

```
----- SUPERCE UTILITY -----
OPTION ===> _

blank - Compare Data Sets          P - Select Process Options
B   - Submit Batch Data Set Compare E - Edit Statements Data Set
S   - Extended Search-For Compare Utility A - Activate/Create Profiles

New DS Name   ===>
Old DS Name   ===>
PDS Member List ===> (blank/pattern - member list, * - compare all)
                    (Leave New/Old Dsn "blank" for concatenated-uncataloged-password panel)
Optional Section
Compare Type  ===> LINE              (FILE/ LINE /WORD/BYTE)
Listing Type  ===> DELTA             (OVSUM/ DELTA /CHNG/LONG/NOLIST)
Listing Dsn   ===> SUPERC.LIST
Process Options ===>
Statements Dsn ===> SUPERC.STMTS
Update Dsn    ===>
BROWSE Output ===> YES              ( YES /NO/COND/UPD)
```

Figure 83. SuperCE Utility Panel

A default compare type, listing type, listing data set name, and Browse option are provided if you choose not to specify your own. The fields on the SuperCE Utility panel are:

New DS Name and Old DS Name

Specify the name of a sequential data set, PDS, or membered PDS. Use standard TSO naming conventions, including quotes for fully qualified names. Leave either or both of these fields blank to display a panel on which you can specify concatenated, uncataloged, and password-protected data sets. These panels are shown in Figure 84 on page 187 (foreground compare) and Figure 85 on page 188 (batch compare).

PDS Member List

Leave this field blank to display a member selection list for the “new” data set. Otherwise, enter one of the following. See “SuperC Member Lists” on page 176 if you need more information.

pattern Entering a pattern causes PDF to display a list of the members in the “new” data set that match the pattern. See “Displaying Member Lists” on page 38 for more information about using patterns. For example:

```
PDS Member List ===>
```

***** Entering an asterisk causes all the members in the “new” data set to be compared to any like-named members in the “old” data set. A member list is *not* displayed. For example:

```
PDS Member List ===> *
```

Members in either data set not having like-named members in the other data set are not compared, but are listed in the output list data set.

Note: You can also use SELECT process statements in the statements data set to specify an optional set of PDS members to be searched. However, the SELECT statement turns off the PDS member list function.

Compare Type

The type of comparison you want SuperC to perform. Enter one of the following:

FILE Compares source data sets for differences, but does not show what the differences are. This is the simplest and fastest method with the least amount of processing overhead. For this compare type, SuperC prepares summary information only and causes all listing types to produce the same output, except NOLIST, which does not produce any output listing. A message is returned to notify you of the compare results.

LINE Compares source data sets for line differences. This compare type is the default. It is most useful for comparisons of program source code because it is record-oriented and points out inserted or deleted lines of code. Lines can be of unlimited size.

WORD Compares source data sets for word differences. In this context, a “word” is a group of characters that begins and ends with a blank or other line delimiter. If you use the XWDCMP process option, all non-alphanumeric characters are considered to

be delimiters. Also, a word cannot be longer than 256 characters.

The WORD compare type is most useful for comparing text data sets. If two data sets contain the same words in the same order, SuperC considers them to be identical, even if those words are not on the same lines.

BYTE Compares source data sets for byte differences. The output listing data set consists of a hexadecimal printout with character equivalents listed on the right. A BYTE compare with a LONG listing of a data set against itself results in a hexadecimal dump of that data set. This compare type is most useful for comparing machine readable data.

Listing Type

The type of listing you want SuperC to create when it compares the data sets. Listing Type is not a required field in SuperCE. If you do not specify a listing type, the default is DELTA. See "SuperC Listings" on page 344 for sample listings.

DELTA Lists the differences between the source data sets, followed by the general summary.

CHNG Lists the differences between the source data sets, plus up to 10 matching lines before and after the differences. This listing is a variation of the DELTA listing; the matching lines before and after help you recognize changed areas of the source data sets.

LONG Lists all the "new" data set source lines, plus "old" data set deleted lines. Both inserted and deleted lines are flagged.

OVSUM Lists only the general summary of the comparison. However, a PDS comparison generates an individual summary line for each PDS member.

NOLIST Produces no listing output. In foreground mode, a message is returned to show the outcome of the comparison.

Listing Dsn

The name of the list data set to which SuperC writes the results of the comparison. However, if you enter NOLIST in the Listing Type field, SuperC does not create an output listing, so this name is ignored.

If you leave this field blank, SuperC allocates a list data set, using default data set attributes and the following data set name:

`prefix.userid.SUPERC.LIST`

where “prefix” is your TSO prefix and “userid” is your user ID. If your prefix and user ID are identical, only your prefix is used. Also, if you do not have a prefix, only your user ID is used.

If you enter the name of a data set that already exists, the contents of that data set is replaced by the new output listing. However, if the data set is sequential, you can add this listing to the data set instead of replacing it by using the APNDLST process option.

If you enter the name of a data set that does not exist, SuperC allocates it for you. The data set is allocated as a sequential data set unless you enter a member name after it, in which case it is allocated as a member of a partitioned data set.

Process Options

Keywords that tell SuperC how to process the compare operation. You can type these keywords in the Process Options fields or select them from a panel. See “P - Select SuperCE Process Options” on page 194 for a table of keywords.

Statements Dsn

The name of the data set that contains your process statements. All statements data sets must be fixed block with 80-byte records (FB 80). See “E - Edit SuperCE Statements Data Set” on page 194 if you need more information.

Update Dsn

Tells SuperC the name of the data set that will contain column-oriented results of the comparison. This data set is normally used as input to post processing programs and can be specified besides the normal listing data set.

If you leave this field blank and use an update (UPDxxxx) option, SuperC uses the following default name:

`prefix.userid.SUPERC.UPDATE`

where “prefix” is your TSO prefix and “userid” is your user ID. If your prefix and user ID are identical, only your prefix is used. Also, if you do not have a prefix, only your user ID is used.

If you enter the name of a data set that already exists, the contents of that data set are replaced by the new update output. However, if the data set is sequential, you can add this listing to the data set instead of replacing it by using the APNDUPD process option.

If you enter the name of a data set that does not exist, SuperC allocates it for you. The data set is allocated as a sequential data

SuperCE Utility (Option 3.13)

set unless you enter a member name after it, in which case it is allocated as a partitioned data set.

Note: For the UPDMVS8, UPDCMS8, UPDSEQ0 and UPDPDEL process options, the update data set contains valid data, but only after a successful compare when differences are detected. The data set is always empty after a comparison that shows the data sets or members being compared have no differences.

BROWSE Output

Tells PDF whether you want to display the output listing in Browse mode. Enter one of the following:

Note: The NOLIST listing type overrides YES, NO, and COND.

YES	Invoke Browse to display the listing data set after processing the comparison. This is the default.
NO	Do not invoke Browse to display the SuperC listing data set.
COND	Do not invoke Browse unless SuperC finds differences between the data sets.
UPD	Browse the update data set instead of the list data set. This parameter is not valid unless you create an update data set by using one or more of the SuperC process options that begin with UPD (UPDxxxx).

The following sections describe the options shown at the top of the SuperCE Utility panel.

Blank - Compare Data Sets

If you leave the OPTION field blank, SuperC processes the data sets in foreground mode, so you can browse the results of the compare. This choice locks your keyboard until SuperC processing is complete.

The panel shown in Figure 84 on page 187 is displayed if you leave the OPTION field and the New DS Name or Old DS Name field blank on the SuperCE Utility panel.

For fields DS1 through DS4, both "NEW" and "OLD," use normal TSO naming conventions. You can specify a series of concatenated data sets, an uncataloged or password-protected data set, or a cataloged data set.

Up to four data sets can be concatenated for both the "NEW" and "OLD" input data sets. Make sure the data sets are concatenated in the proper sequence, as follows:

1. If two or more sequential data sets are concatenated as one input data set, the data set attributes, such as block size, must be identical.

```

----- SUPERCE - CONCATENATION INTERACTIVE ENTRY -----
COMMAND ===> _

"NEW"      DS1 ===>
CONCATENATION DS2 ===>
           DS3 ===>
           DS4 ===>
OTHER "NEW" PARTITIONED OR SEQUENTIAL DATA SET
  DATA SET NAME ===>
  VOLUME SERIAL  ===> (If not cataloged)
  PASSWORD       ===> (If password protected)

"OLD"      DS1 ===>
CONCATENATION DS2 ===>
           DS3 ===>
           DS4 ===>
OTHER "OLD" PARTITIONED OR SEQUENTIAL DATA SET
  DATA SET NAME ===>
  VOLUME SERIAL  ===> (If not cataloged)
  PASSWORD       ===> (If password protected)

```

Figure 84. SuperCE - Concatenation Interactive Entry Panel

2. PDS concatenations must have the data set with the largest block size as the first in any concatenation.
3. SuperCE uses only the first occurrence of a member in the concatenated series of PDSs as source input for a compare. Any other occurrences of the member are ignored.

“Other” partitioned or sequential data sets, volume serials, and data set passwords are specified as on any other data entry panel. See Chapter 2, “Libraries and Data Sets” if you need more information.

Note: The PASSWORD fields apply only to the “other” partitioned or sequential data sets. TSO prompts you if any concatenated data sets are password-protected.

B - Submit Batch Data Set Compare

Option B causes SuperC to process the data sets in batch mode. This choice frees the keyboard, allowing you to continue using PDF while waiting for SuperC to compare the data sets. The output listing is sent to the destination specified on the SuperC Utility - Submit Batch Jobs panel (Figure 81 on page 178).

The panel shown in Figure 85 on page 188 is displayed if you select option B and leave the New DS Name or Old DS Name field blank on the SuperCE Utility panel. You can concatenate up to four data sets that have like attributes. For example, all must be either sequential or partitioned.

SuperCE Utility (Option 3.13)

```
----- SUPERCE - CONCATENATION BATCH ENTRY -----
COMMAND ---> _

"NEW"      DS1 --->
CONCATENATION DS2 --->
              DS3 --->
              DS4 --->
OTHER "NEW" PARTITIONED OR SEQUENTIAL DATA SET
  DATA SET NAME --->
  VOLUME SERIAL --->                                (If not cataloged)

"OLD"      DS1 --->
CONCATENATION DS2 --->
              DS3 --->
              DS4 --->
OTHER "OLD" PARTITIONED OR SEQUENTIAL DATA SET
  DATA SET NAME --->
  VOLUME SERIAL --->                                (If not cataloged)
```

Figure 85. SuperCE - Concatenation Batch Entry Panel

This panel is the same as the panel shown in Figure 84 on page 187, except the PASSWORD field is omitted. If your data sets are password-protected, compare the data sets in foreground mode by leaving the OPTION field on the SuperCE Utility panel blank.

Printing a SuperCE Listing in Batch Mode

If you selected option B on the SuperCE Utility panel, the panel shown in Figure 81 on page 178 is displayed before the job is submitted. This panel lets you determine whether to print your SuperC listing or write it to a list data set.

S - Extended Search-For Compare Utility

Note: For an introduction to the Search-For and Extended Search-For utilities (options 3.14 and 3.13.S), see Appendix E, "SuperC Reference" on page 367.

If you select option S, the Extended Search-For Utility panel, shown in Figure 86 on page 189, is displayed. This utility is a dialog that uses the SuperC program to search your data sets or PDS members for one or more character strings. It is appropriate if you need more flexibility than the standard Search-For utility (3.14) provides.

```

----- EXTENDED SEARCH-FOR UTILITY -----
OPTION ==> _

blank - Search-For Strings                P - Select Process Options
                                           E - Edit Statements Data Set

Search DS Name ==>
PDS Member List ==> (blank/pattern - member list, * - search all)
(Leave Search Dsn "blank" for concatenated-uncataloged-password panel)

Enter Search Strings and Optional operands (WORD/PREFIX/SUFFIX,C)

CAPS ---->
CAPS ---->
CAPS ---->
ASIS ---->
ASIS ---->

Listing DS Name ----> SRCHFOR.LIST
Process Options ---->
Statements Dsn ----> SRCHFOR.STMTS

```

Figure 86. Extended Search-For Utility Panel

The character string(s) is the only entry the panel requires. The fields on this panel are:

Search DS Name

Specify the name of a sequential data set, PDS, or membered PDS. Use standard TSO naming conventions, including quotes for fully qualified names. Leave this field blank to display a panel on which you can specify concatenated, uncataloged, and password-protected data sets. This panel is shown Figure 87 on page 192.

PDS Member List

Leave this field blank to display a list of all the members in the search data set. Otherwise, enter one of the following. See "Search-For Member Lists" on page 202 if you need more information.

pattern

Entering a pattern causes PDF to display a list of the members in the search data set that match the pattern. See "Displaying Member Lists" on page 38 for more information about using patterns. For example:

```
PDS Member List ==> ISR*
```

*

Entering an asterisk causes all the members in the search data set to be searched. For example:

```
PDS Member List ==> *
```

Note: You can also use SELECT process statements in the statements data set to specify an optional set of PDS members to be searched. However, the SELECT statement turns off the PDS member list function.

SuperCE Utility (Option 3.13)

CAPS A search string that you want the Extended Search-For utility to find. This search string is converted to uppercase before the search begins and is found only if it exists in the search data set in uppercase.

Note: The ANYC process option causes the string to be found in any case, (upper, lower, or mixed) even if you enter the string in the CAPS field.

You can enter up to three uppercase search strings, one in each CAPS field. Here are some examples:

- Either of the following strings may be found in the search data set:

```
CAPS ==> THEN
CAPS ==> ISR
```

- The following two strings must be found on the same line because of the continuation (C) keyword. THEN must be a complete word, while ISR must be the prefix of a word.

```
CAPS ==> THEN CAP
CAPS ==> ISR PREFIX
```

- In the next example, a hexadecimal string is specified as the search string. Use this to find unprintable characters.

```
CAPS ==> A1234567
```

- The following example searches for the string JOE'S CLIST. Notice that the string is enclosed in single quotation marks and the apostrophe following Joe's name has been doubled.

```
CAPS ==> 'JOE'S CLIST'
```

ASIS A search string that you want the Extended Search-For utility to find. This search string is searched for on an "as is" basis. The Extended Search-For utility does not find the string unless it exists in the data set exactly as you enter it in an ASIS field. You can enter up to two "as is" search strings, one in each ASIS field.

The examples following the CAPS field definition apply to the ASIS field as well.

Notes:

1. See "Search-For Strings and Keywords" on page 201 for a list of rules that determine the format required for entering search strings and for definitions of the keywords that are shown in the examples below.
2. The SRCHFOR and SRCHFORC process statements override any strings entered in the CAPS and ASIS fields.

Listing DS Name

The name of the list data set to which the Extended Search-For utility writes the listing information. If you leave this field blank, Extended Search-For allocates a list data set, using default data set attributes and the following data set name:

```
prefix.userid.SRCHFOR.LIST
```

where “prefix” is your TSO prefix and “userid” is your user ID. If your prefix and user ID are identical, only your prefix is used. Also, if you do not have a prefix, only your user ID is used.

If you enter the name of a data set that already exists, the contents of that data set are replaced by the new listing output. However, if the data set is sequential, you can add this listing to the data set instead of replacing it by using the APNDLST process option.

If you enter the name of a data set that does not exist, Search-For allocates it for you. The data set is allocated as a sequential data set unless you enter a member name after it, in which case it is allocated as a partitioned data set.

Process Options

Keywords that tell SuperC how to process the search-for operation. You can type these keywords in the Process Options field or select them from a panel. See “Process Options” on page 374 for tables of keywords.

Statements Dsn

The name of the data set that contains your search-for process statements, which you can create or change by using option E on the Extended Search-For Utility panel. SuperC reads these process statements before conducting the search. All statements data sets must be fixed block with 80-byte records (FB 80).

The following sections describe the options shown on the Extended Search-For Utility panel.

Blank - Search-For Strings

If you leave the Search DS Name field blank, the Extended Search-For - Concatenation Data Set Entry panel, shown in Figure 87 on page 192, is displayed.

For fields DS1 through DS4, use normal TSO naming conventions. You can specify a series of concatenated data sets, an uncataloged or password-protected data set, or a cataloged data set name.

Up to four data sets can be concatenated. Make sure the data sets are concatenated in the proper sequence, as follows:

1. If two or more sequential data sets are concatenated as one input data set, the data set attributes, such as block size, must be identical.


```
----- EXTENDED SEARCH-FOR - CONCATENATION DATA SET ENTRY -----
COMMAND ----> _

"SEARCH"      DS1 ---->
CONCATENATION DS2 ---->
              DS3 ---->
              DS4 ---->

OTHER "SEARCH" PARTITIONED OR SEQUENTIAL DATA SET
DATA SET NAME ---->
VOLUME SERIAL ---->                (If not cataloged)
PASSWORD       ---->                (If password protected)
```

Figure 87. Extended Search-For - Concatenation Data Set Entry Panel

2. PDS concatenations must have the data set with the largest block size as the first in any concatenation.
3. Search-For uses only the first occurrence of a member in the concatenated series of PDSs as source input for a search. Any other occurrences of the member are ignored.

“Other” partitioned or sequential data sets, volume serials, and data set passwords are specified as on any other data entry panel. See Chapter 2, “Libraries and Data Sets” if you need more information.

Note: The PASSWORD field applies only to the “other” partitioned or sequential data set. TSO prompts you if any concatenated data sets are password-protected.

P - Select Search-For Process Options

The following rules govern the selection of Search-For process options:

- Type any non-blank character to the left of a process option(s). Then press the ENTER key. This causes the option(s) you chose to be displayed in the Process Options field on the Extended Search-For Utility panel. If you select two options that cannot be chosen together, or if you enter an option name incorrectly, an error message is displayed.
- Enter the CANCEL command to return to the Extended Search-For Utility panel without processing selections.

Search-For process options can affect how the input data is processed, and determine the format and content of the output listing data set. They can also help you save processing time by avoiding comments and blank lines.

NW: 15457 DocId: 70002077 Page 579 All process options except the “do not process” (DPxxxx) options can be chosen from the Search-For Process Options panel. However, you can type

any of them in the Process Options field on the Extended Search-For Utility panel. Errors caused by mistyping process options are detected when you invoke the Extended Search-For utility.

For definitions of the Search-For process options, see "Process Options" on page 378.

E - Edit Search-For Statements Data Set

A statements data set consists of process statements that contain instructions for the SuperC program. They are similar to the process options, but are composed of a keyword and one or more operands. See "P - Select Search-For Process Options" on page 192 for information about Search-For process options.

When you select option E from the Extended Search-For Utility panel, the Extended Search-For utility displays the statements data set you specified in the Statements Dsn field. This data set is always displayed in Edit mode, allowing you to add, change, or delete search-for process statements as needed.

The size of the Edit window depends on the number of lines your terminal can display. The sample panel shown in Figure 88 shows how the Edit window appears on a 24-line display. Examples of some common process statements are listed below the Edit window so you can easily compose the proper input line.

```

EDIT --- FBROWN.SRCHFOR.STMTS ----- COLUMNS 001 072
COMMAND ==> _                               SCROLL ==> PAGE

      Enter or change Process Statements in the EDIT window below:
***** ***** TOP OF DATA *****
.....
.....
.....
.....
***** ***** BOTTOM OF DATA *****
Examples      Explanation
SRCHFOR 'ABCD',W      Search for the word "ABCD"
SRCHFORC 'DEFG'      "DEFG" must be on same line as word "ABCD"
CMPCOLM 1:60 75:90    Search columns 1:60 and 75:90 for string(s)
DPLINE 'PAGE ',87:95  Exclude line if "PAGE " found in columns 87:99
DPLINE 'PAGE '        Exclude if "PAGE " found anywhere on line
SELECT MEM1,MEM2      Search only members MEM1 and MEM2 of PDS
Others: DPLINEC LSTCOLM CMPLINE NTITLE NCHGT LNCT SLIST
        comment-lines ("*" and ".,*")
  
```

Figure 88. Search-For Process Statements Panel

The SuperC program validates the process statements at run time. Invalid process statements are not used and are noted at the bottom of the listing. Unless a higher return code is required by some other condition, a return code of 4 is returned.

For the syntax and examples of the Search-For process statements, see "Process Statements" on page 393.

SuperCE Utility (Option 3.13)

P - Select SuperCE Process Options

To select one or more SuperCE process options, do either of the following:

- Type any non-blank character to the left of a process option(s). If “continued on next page” is shown at the bottom of the panel, you can press the ENTER key to display more process options. Enter the END command when you finish selecting options. This causes the options you chose to be displayed in the Process Options field(s) on the SuperCE Utility panel. If you select two options that cannot be chosen together, or if you enter an option name incorrectly, an error message is displayed.
- Enter the CANCEL command to return to the SuperCE Utility panel without processing selections.

SuperC process options can affect how the input data is processed, and determine the format and content of the output listing data set. They can also help you save processing time by avoiding comments and blank lines. A separate group of options, called update data set options (UPDxxxx), let you create update data sets, examples of which are shown in “Update Data Set Control Options” on page 382.

All these options can be chosen from the XXXX Compare Process Options panels, where XXXX is the compare type (FILE, LINE, WORD, or BYTE) that you are using. The only exception is that the “do not process” (DPxxxx) options are not available from a selection panel when you are conducting a WORD comparison. However, you can type any of them in the Process Options field on the SuperCE Utility panel. Errors caused by mistyping process options are detected when you invoke the SuperCE utility.

For definitions of the SuperC process options, see “Process Options” on page 378.

E - Edit SuperCE Statements Data Set

A statements data set consists of process statements that contain instructions for the SuperC program. They are similar to the process options, but are composed of a keyword and one or more operands. See “P - Select SuperCE Process Options” for information about SuperCE process options.

When you select option E from the SuperCE Utility panel, the SuperCE utility displays the statements data set you specified in the Statements Dsn field. This data set is always displayed in Edit mode, allowing you to add, change, or delete SuperC process statements as needed. Only one process statement can appear on each line of the statements data set.

The size of the Edit window depends on the number of lines your terminal can display. The sample panel shown in Figure 89 on page 195 shows how the Edit window appears on a 24-line display. Examples of some common process statements are listed below the Edit window so you can easily compose the proper input line.

```

EDIT --- FBROWN.SUPERC.STMTS ----- COLUMNS 001 072
COMMAND ----> _                      SCROLL ----> PAGE

Enter or change Process Statements in the EDIT window below:
***** TOP OF DATA *****
.....
.....
.....
.....
.....
***** BOTTOM OF DATA *****
Examples      Explanation
CMPCOLM 5:60 75:90  Compare using two column compare ranges
LSTCOLM 25:90      List columns 25:90 from input
DPLINE 'PAGE '      Exclude line if "PAGE " found anywhere on line
DPLINE 'PAGE ',87:95 Exclude if "PAGE " found within columns 87:99
SELECT MEM1,NMEM2:OMEM2 Compare MEM1 with MEM1 and NMEM2 with OMEM2
CMPLINE NTOP 'MACRO' Start comparing after string found in new DSN
LNCT 66            Set lines per page to 66
Others: DPLINEC CMPBOFS CMPCOLMN CMPCOLMO NTITLE OTITLE SLIST
        NCHGT  OCHGT  comment-lines ("*" ".")

```

Figure 89. SuperC Process Statements Panel

The SuperC program validates process statements at run time. Invalid process statements are not used and are noted at the bottom of the listing. Unless a higher return code is required by some other condition, a return code of 4 is generated.

See "Process Statements" on page 393 for process statement syntax, definitions, and examples.

A - Activate/Create Profiles

A SuperC profile is a data set that can contain a compare type, a listing type, a Browse setting, and various combinations of process options and process statements that you select.

SuperC profiles are useful for a wide range of users. Beginners can use profiles created by others as a simple method of running SuperC. Experienced SuperC users can create profiles for the groups of options they use often so that they do not have to remember individual process options and statements. Also, profiles give system programmers a mechanism for setting up complex compare tools that others can simply invoke by profile name.

Some other characteristics of profiles are:

- A profile can be either a sequential data set or a member of a PDS.
- Data set names are not represented in a profile.
- Profiles can be created only with the SuperCE utility (option 3.13). However, once they are created, they can be used in the standard SuperC utility (option 3.12).

NW: 15457 DocId: 70002077 Page 582 • To change a profile, activate it and make the necessary changes to the information in the fields on the SuperCE Utility panel. Then select

SuperCE Utility (Option 3.13)

option C and create a profile, entering in the Activate/Create Profile DS Name field the name of the profile data set that you want to modify.

- You can display the contents of a profile data set by using the Browse and Edit options. Figure 90 shows a Browse display of a profile data set.

```
BROWSE -- FBROWN.SUPERC.PROFILE ----- LINE 000000 COL 001 080
COMMAND ===> _ SCROLL ===> PAGE
***** TOP OF DATA *****
.* PROF PREFIX CTYPE=LINE,LTYPE=DELTA ,BRW=YES
.* PROF PREFIX PROC1=NOSEQ DPPLCMT DPADCMT DPFTCMT DPCBCMT * MARGIN*
.* PROF PREFIX PROC2=DPBLKCL LOCS CNPML WIDE UPDCNTL * MARGIN*
SELECT ISRUC1A,ISRUC1B,ISRUC2:ISRUC3 00000100
CMPCOLM 7:60 00000200
DPLINE 'EXAMPLE',7:60 00000300
LNCT 65 00000400
***** BOTTOM OF DATA *****
```

Figure 90. SuperCE Profile Browse Display

Selecting option A on the SuperCE Utility panel displays the panel shown in Figure 91. This panel is used to activate and create profiles.

```
----- SUPERCE - PROFILE MANAGER -----
OPTION ===> _

A - Activate - Reads the specified input profile data set:
               1. Establishes the process and compare options from the profile
                  prefix lines.
               2. Establishes the profile as the process statement data set if
                  any process statements are detected.

C - Create - Creates an output profile data set:
              1. Combines process and compare options from the Primary Panel
                  and any process statements from the Statements Data Set
                  SUPERC.STMTS
              2. Rewrites the profile data set (if the data set exists) or
                  allocates a new data set before generating the profile.

Activate/Create
Profile DS Name ===>
```

Figure 91. SuperCE - Profile Manager Panel

The only field on this panel is:

Activate/Create Profile DS Name

The name of the profile data set that you want to either activate or create. This field is required.

If you choose option A (Activate), SuperCE copies the compare type, listing type, Browse setting, and any process options this data set contains. If the profile also contains process statements, SuperCE displays the name of the profile data set in the Statements Dsn field on the SuperCE Utility panel.

If you choose option C (Create), SuperCE copies the information that is saved into one output data set or member.

The following sections describe the options shown at the top of the SuperCE - Profile Manager panel.

A - Activate

Option A (Activate) restores the contents of the profile data set specified in the Activate/Create Profile DS Name field. This data set must be cataloged. When you press the ENTER key, information that was stored in the profile data set when it was created is displayed in the corresponding fields on the SuperCE Utility panel. For example, process options stored in the profile appear in the Process Options fields. At this point, you can make additional changes or choose another option.

C - Create

Option C (Create) causes SuperCE to copy data entered on the SuperCE Utility panel and place it in the profile data set specified in the Activate/Create Profile DS Name field. Be sure the correct information is displayed on that panel and that the statements data set, if you specify one, contains the correct process statements before you create the profile.

If the profile data set that you specify does not already exist, SuperCE allocates it for you. Data stored in the profile data set can include:

- The following values taken from the fields on the SuperCE Utility panel. The abbreviations in parentheses show how these values are identified in a profile data set:
 - Compare type (CTYP)
 - Listing type (LTYP)
 - Browse setting (BRW)
 - Process options (PROC1 and PROC2).
- Process statements copied from the statements data set that was specified in the Statements Dsn field. This data set name is displayed and highlighted on the SuperCE - Profile Manager panel. For example, the sample panel shown in Figure 91 on page 196 displays the name SUPERC.STMTS.

SuperCE Utility (Option 3.13)

If you leave the Statements Dsn field blank, the data set name and the words preceding it are not displayed on the SuperCE - Profile Manager panel and SuperCE does not include any process statements in your profile. See the following sections for information about process options and process statements, respectively:

- “P - Select SuperCE Process Options” on page 194
- “E - Edit SuperCE Statements Data Set” on page 194.

Search-For Utility (Option 3.14)

Note: For an introduction to the Search-For and Extended Search-For utilities (options 3.14 and 3.13.S), see Appendix E, "SuperC Reference" on page 367.

The Search-For utility (3.14) is a dialog that uses the SuperC program to search your data sets or PDS members for one or more character strings. The Search-For Utility panel, shown in Figure 92, is the first panel of the Search-For utility. The only requirements for this panel are:

- A string to be searched for, unless you enter YES for multiple strings
- A data set to search, along with a volume serial and password if necessary.

A default listing data set name is provided if you choose not to enter your own.

```

----- SEARCH-FOR UTILITY -----
COMMAND ---->

SEARCH STRING
----> _

MULTIPLE STRINGS ----> yes (Yes to specify additional search strings)

ISPF LIBRARY:
PROJECT ----> FBROWN
GROUP ----> PRIVATE ----> ----> ---->
TYPE ----> TABLES
MEMBER ----> (Blank or pattern for member selection list,
              '*' for all members)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ---->
VOLUME SERIAL ----> (If not cataloged)

DATA SET PASSWORD ----> (If password protected)

LISTING DSNAME ----> SRCHF0R.LIST
  
```

Figure 92. Search-For Utility Panel

All the fields on this panel are explained in Chapter 2, "Libraries and Data Sets," except the following:

SEARCH STRING

A string to be searched for. No distinction is made between uppercase and lowercase characters. Use the Extended Search-For utility (option 3.13.S) to specify case-sensitive searches.

Four keywords—C, PREFIX, SUFFIX, and WORD—can help you narrow the scope of a search. See "Search-For Strings and Keywords" on page 201 for information about these keywords and the rules that govern search string entry.

Search-For Utility (Option 3.14)

MULTIPLE STRINGS

Tells Search-For whether you want to search for more than one string. The valid parameters are:

blank or NO

Search for the string entered in the SEARCH STRING field only. This makes SEARCH STRING a required field, because the panel that lets you enter multiple strings is not displayed.

YES

Search for more than one string. This value causes the Search-For utility to display the panel shown in Figure 93 on page 201, on which you can specify additional search strings. This panel precedes a member list request.

LISTING DSNAME

The name of the listing data set to which the SuperC program writes the results of the search. If you leave this field blank, Search-For allocates a listing data set, using default data set attributes and the following data set name:

```
prefix.userid.SRCHF0R.LIST
```

where “prefix” is your TSO prefix and “userid” is your user ID. If your prefix and user ID are identical, only your prefix is used. Also, if you do not have a prefix, only your user ID is used.

If you enter the name of a data set that does not exist, Search-For allocates it for you. The data set is allocated as a sequential data set unless you enter a member name after it, in which case it is allocated as a partitioned data set.

Entering Multiple Search Strings

The panel shown in Figure 93 on page 201 is displayed if you enter YES in the MULTIPLE STRINGS field on the Search-For Utility panel (Figure 92 on page 199). You can specify:

- Additional strings to be searched for
- Optional scan-type and continuation keywords.

From this panel, pressing the ENTER key either:

- Displays a member list, if requested
- Runs the search if no member list is needed.

Entering the END command returns you to the Search-For Utility panel.

SEARCH --- FBROWN.PRIVATE.TABLES -----
COMMAND ---->

Specify 1 or more SEARCH STRINGS below:

----> _

---->

---->

---->

---->

---->

---->

---->

---->

---->

Press ENTER to start search or END command to exit.

Figure 93. Multiple Search Strings Panel

Search-For Strings and Keywords

Enter the string(s) that you want SuperC to find. When you press the ENTER key, SuperC looks for the strings without regard to whether they appear in uppercase or lowercase in the original data set. If you want the search to distinguish between uppercase and lowercase, you must use the Extended Search-For utility (option 3.13.S).

Entering Search Strings

Enclose the string in single quotation marks if it contains imbedded blanks or apostrophes. Two consecutive apostrophes must be entered to specify a single apostrophe within a search string.

The following example searches for the string "IT'S A LIVING."

```
==> 'IT'S A LIVING.'
```

Using Keywords

The following keywords can help you narrow the range of the search. If you do not use a keyword, SuperC will find the string wherever it exists, even if that happens to be in the middle of a word.

PREFIX Shows the string is preceded by a non-alphanumeric character, such as a blank space. It cannot be used on the same line with SUFFIX or WORD. For example, you can do this:

```
==> PREFIX 'IT'S A LIVING.'
==> PREFIX 'IT'S A LIVING.'
```

Search-For Utility (Option 3.14)

but not this:

```
==> ELSE PREFIX SUFFIX
```

SUFFIX Shows the string is followed by a non-alphanumeric character. It cannot be used on the same line with PREFIX or WORD. See the examples under PREFIX.

WORD Shows the string is preceded *and* followed by a non-alphanumeric character. It cannot be used on the same line with PREFIX or SUFFIX. See the examples under PREFIX.

C Continuation. Shows continuation of the previous line(s). Continuation lines generate additional strings, *all* of which must be found in the same line of an input data set.

Also, the C keyword can be entered on the same line as one of the other keywords. The following example tells SuperC to find ELSE and to also find IF, but only when IF is on the same line as ELSE.

```
==>  
==>
```

Search-For Member Lists

A panel similar to the one shown in Figure 94 on page 203 is displayed only if:

- The search data set(s) is partitioned.
- The MEMBER field on the Search-For Utility panel (Figure 92 on page 199) or the PDS Member List field on the Extended Search-For Utility panel (Figure 86 on page 189) was left blank or a pattern was used. See “Displaying Member Lists” on page 38 for more information about displaying member lists.
- For the Extended Search-For utility (option 3.13.S), the statements data set being used does not contain any SELECT process statements.

To start the search, enter the END command.

To cancel your selections, enter either:

- The RESET command to remove all unprocessed selections without ending the member list display
- The CANCEL command to end the member list display without processing selections still on the screen.

Note: Both the jump function (=) and the RETURN command cause an implied cancellation of selections before they are implemented.

SEARCH -- FBROWN.PRIVATE.TABLES ----- ROW 00001 OF 00008
COMMAND ----> _ SCROLL ----> PAGE

Enter END command to process selections or CANCEL to exit the member list.

NAME	VV.MM	CREATED	CHANGED	SIZE	INIT	MOD	ID
ISRUC1A	01.28	86/01/22	86/03/10 14:15	24	24	0	FBROWN
ISRUC1B	01.27	86/01/22	86/03/10 14:40	24	21	3	FBROWN
ISRUC1C	01.10	86/01/30	86/03/10 14:40	24	25	4	FBROWN
ISRUC1S	01.11	86/02/19	86/03/10 14:51	23	17	2	FBROWN
ISRUC1T	01.24	86/01/30	86/03/10 16:21	25	15	5	FBROWN
ISRUC2	01.28	86/02/13	86/03/10 14:30	24	20	2	FBROWN
ISRUC6L	01.10	86/02/25	86/03/10 12:19	24	21	1	FBROWN
ISRUC7	22.09	86/02/07	86/03/10 12:06	21	39	3	FBROWN

END

Figure 94. Search Member List Panel

Search-For Utility (Option 3.14)

CHAPTER 7. FOREGROUND (Option 4)

This chapter describes Foreground, option 4 on the ISPF/PDF Primary Option Menu. The Foreground option provides an interface to:

- **Standard language processors for foreground compilation and assembly of programs stored in ISPF libraries or TSO data sets**
- **The linkage editor**
- **The Document Composition Facility (SCRIPT/VS) program product for formatting, browsing, and printing documents**
- **The FORTRAN and COBOL interactive debug facilities**
- **The member parts list, a facility for listing the parts for a source program module.**

This chapter describes the foreground processing for each of these interfaces.

Chapter 7. Foreground (Option 4)

The Foreground option (4) allows PDF to run the foreground processors shown on the Foreground Selection Panel, Figure 95. All these processors, except SCRIPT/VS, COBOL interactive debug, and FORTRAN interactive debug, are also available with the Batch option (5).

When you run a foreground processor, you must wait until the processor ends before doing anything else with PDF. However, if you want to use PDF while waiting for the processor to end, submit the input as a batch job. You can do this by using the Batch option if the processor you need is listed on the Batch Selection Panel, Figure 129 on page 251.

```

----- FOREGROUND SELECTION PANEL -----
OPTION ===> _

1 - Assembler H              6 - PASCAL/VS compiler
1A - Assembler XF            7 - Linkage editor
2 - VS COBOL II compiler      9 - SCRIPT/VS
2A - OS/VS COBOL compiler    10 - VS COBOL II interactive debug
3 - VS FORTRAN compiler      10A - COBOL interactive debug
4 - PL/I checkout compiler   11 - FORTRAN interactive debug
5 - PL/I optimizing compiler 12 - Member parts list

SOURCE DATA PACKED      ===> NO      (YES or NO)
ENTER SESSION MANAGER MODE ===>      (YES or NO)
  
```

Figure 95. Foreground Selection Panel

Foreground Processing Sequence

The main foreground processing sequence is:

1. Decide which foreground processor you want to use, keeping the following requirements in mind:
 - To compile a program by using the OS/VS COBOL compiler (option 4.2A), you must use the Data Set utility (option 3.2) to allocate a symbolic debug data set and, optionally, a print output data set. You will use these data sets later if you need to correct your program by using COBOL interactive debug (option 4.10A). Allocate these data sets and then continue with the next step. See

Foreground Processing Sequence

“Symbolic Debug Data Sets” on page 242 and “Print Output Data Sets” on page 243 if you need more information.

- If you have compiled a program by using the OS/VS COBOL compiler and now want to run COBOL interactive debug, you must first use the linkage editor (option 4.7 or option 5.7) to generate a load module.

When you have satisfied these requirements or chosen a different processor, continue with the next step.

2. If you do not know whether the source data is in packed format, find out by editing the data set and entering the PROFILE command. The profile shows either PACK ON (data is stored in packed format) or PACK OFF (data is not stored in packed format).

If the data is not packed, enter NO in the SOURCE DATA PACKED field. If the data is packed, enter YES.

Also, you should read “Expanding Packed Data” on page 211, paying close attention to:

- Information that applies to the foreground processor you plan to use
- The difference between expanding a sequential data set and expanding members of a partitioned data set.

When you are satisfied that the data set is ready to be processed, continue with the next step.

3. Select one of the foreground processors listed at the top of the Foreground Selection Panel (Figure 95 on page 205).

Note: A region size of two megabytes or more will probably be required to run the VS FORTRAN compiler in the foreground.

4. Enter YES or NO in the SOURCE DATA PACKED field to tell PDF whether it will need to expand the source data.

Note: The SOURCE DATA PACKED field has no effect on the Member parts list option (4.12). Member parts list can read both packed and unpacked data sets, so no expansion is needed.

5. When the Session Manager licensed program, 5740-XE2, is installed, your installation can display a line on the Foreground Selection Panel that allows you to enter Session Manager mode when you invoke any of the foreground processors. Once you invoke Session Manager, it stays in effect for all logical screens until you turn it off. For example, if you invoke Session Manager and then split the screen, Session Manager will be in effect on *both* logical screens.

If the ENTER SESSION MANAGER MODE field is displayed, enter YES or NO to tell PDF whether to invoke Session Manager.

Foreground Processing Sequence

Note: If graphics interface mode is active, Session Manager does not get control of the screen. Graphics interface mode is activated when a GRINIT service has been issued, but a GRTERM service has not been issued. See Dialog Management Services and Examples for more information about these two services.

6. Press the ENTER key.

PDF displays the data entry panel for the processor you selected. The remainder of this processing sequence applies to all foreground processors except SCRIPT/VS, VS COBOL II interactive debug, COBOL interactive debug, and Member Parts List. For these processors, use the sequence referred to in the following list:

Processor	Reference
SCRIPT/VS	"SCRIPT/VS (Option 4.9)" on page 233
VS COBOL II interactive debug	"VS COBOL II Interactive Debug (Option 4.10)" on page 240
COBOL interactive debug	"COBOL Interactive Debug (Option 4.10A)" on page 241
Member parts list	"Member Parts List (Option 4.12)" on page 245

7. Enter the appropriate ISPF library and concatenation sequence or data set name(s). If the input data set is partitioned, you can leave the member name blank or use a pattern to display a member list. If you need help, see:
 - "Naming ISPF Libraries and Data Sets" on page 29 for help in entering library or data set names
 - "Object Data Sets" on page 217 for information about object modules
 - "Input Data Sets" on page 215 for information about the regular concatenation sequence
 - "Linkage Editor Concatenation Sequence" on page 232 for help with the linkage editor concatenation sequence
 - "Displaying Member Lists" on page 38 for information about displaying member lists.

Notes:

- a. *VS FORTRAN has no LIB option, which some foreground processors use to specify the input data set concatenation sequence. Therefore, the concatenation sequence specified in the GROUP fields is used to find the member to be compiled.*

Foreground Processing Sequence

- b. For FORTRAN interactive debug, the TYPE, or last qualifier, must be either OBJ or LOAD. However, if you specify an OBJ data set as your input data set, you must include a load library or data set in the input search sequence (see step 13 on page 209).*
8. This step applies to FORTRAN interactive debug only. Use the SOURCE TYPE field to tell PDF the TYPE, or last qualifier, of the data set used to create the input object module or load module.
9. Use the LIST ID field to tell PDF what to name the output listing. See “List Data Sets” on page 216 if you need more information.
10. This step applies only to the PL/I checkout compiler. In the EXECUTION PARMS field, enter any parameters that you want PDF to pass to the program being compiled. These parameters are the same as those passed to a program with the PARM keyword of an EXEC JCL statement.
11. Enter your password in the PASSWORD field if your input data set is password-protected. See “Password Protection” on page 217 if you need more information.
12. The OPTIONS field, whether ASSEMBLER, COMPILER, LINKAGE EDITOR, or DEBUG, is remembered from one session to another. Therefore, you do not need to change this field unless the options you need are not displayed.

Be careful *not* to enter any options that ISPF generates automatically. These options are listed on the data entry panel. If you need more information about the options available for your processor, go to the page referred to in the following list:

Processor	Reference
Assembler H	“Assembler H (Option 4.1)” on page 219
Assembler XF	“Assembler XF (Option 4.1A)” on page 220
VS COBOL II compiler	“VS COBOL II Compiler (Option 4.2)” on page 221
OS/VS COBOL compiler	“OS/VS COBOL Compiler (Option 4.2A)” on page 223
VS FORTRAN compiler	“VS FORTRAN Compiler (Option 4.3)” on page 225
PL/I checkout compiler	“PL/I Checkout Compiler (Option 4.4)” on page 226
PL/I optimizing compiler	“PL/I Optimizing Compiler (Option 4.5)” on page 228

Foreground Processing Sequence

Pascal/VS compiler	"Pascal/VS Compiler (Option 4.6)" on page 229
Linkage editor	"Linkage Editor (Option 4.7)" on page 231
FORTRAN interactive debug	"FORTRAN Interactive Debug (Option 4.11)" on page 244

13. Enter any additional input libraries you need. For FORTRAN interactive debug, enter any input LOAD libraries that you need to complete the search sequence. These libraries must be LOAD libraries only. See "Input Data Sets" on page 215 if you need help.
14. Once all the input fields have been specified, press the ENTER key to invoke the foreground processor.

If the Session Manager is installed and if you specified Session Manager mode on the Foreground Selection Panel, the foreground processor and all PF and PA keys are under the control of the Session Manager. When foreground processing is complete, you are prompted to enter a null line to return to PDF control.

If the Session Manager is not invoked, the PA and PF keys have their usual TSO-defined meanings; generally, the PF keys are treated the same as the ENTER key.

15. Communication with foreground processors is in line-I/O mode. Whenever you see three asterisks, press the ENTER key.
16. If the foreground processor generated an output listing, the listing is displayed automatically in Browse mode.

Note: If a Foreground processing program ends abnormally, PDF displays a message in the upper-right corner of the screen and does not enter Browse mode. The list data set is retained, but the Foreground Print Options panel (see step 17) is not displayed.

You can scroll the output up or down using the scroll commands. All the Browse commands are available to you. When you finish browsing the listing, enter the END command.

17. An optional print utility exit routine can be installed by your system programmer. If this exit routine is installed, it may cause the Foreground option's response to differ from the following descriptions. See *Installation and Customization* for more information about the print utility exit.

Another factor that can affect the Foreground option's performance is whether the TSO/E Information Center Facility is installed. If the TSO/E Information Center Facility is installed, your installation can optionally allow PDF to display a panel for submitting TSO/E Information Center Facility information with the print request. See Figure 98 on page 218 for an example of this panel and "Using the TSO/E Information Center Facility" on page 153 for information about

Foreground Processing Sequence

the fields on this panel. If the TSO/E Information Center Facility is not installed, the Foreground option displays the panel shown in Figure 96 on page 210 to allow you to print, keep, or delete the output.

```
----- FOREGROUND PRINT OPTIONS -----
OPTION  ===>  _

    PK - Print data set and keep          K - Keep data set (without printing)
    PD - Print data set and delete        D - Delete data set (without printing)

    If END command is entered, data set is kept without printing.

DATA SET NAME: KIRK.LISTASM.LIST

PRINT MODE  ===>                               (BATCH or LOCAL)

SYSOUT CLASS ===> A
PRINTER ID  ===>                               (For 328x printer)

JOB STATEMENT INFORMATION:                    (Required for system printer)
===> //KIRKA JOB (AB01,B001,123456),'M.KIRK'
===> /**
===> /**
===> /**
```

Figure 96. Foreground Print Options Panel

On this panel, the DATA SET NAME field shows the name of the list data set that contains the output generated by the processor you selected. In the OPTION field, enter one of the options shown at the top of the panel. All the options and fields on this panel are described under “Ending PDF or a PDF Function” on page 22, except the PK option. The PK option allows you to prevent the data set from being deleted after it is printed.

When you press the ENTER key, the processor entry panel is displayed again. A message indicating completion of the process is displayed in the upper-right corner of the screen.

18. You can do one of the following:

- Enter other parameters and invoke the same processor.
- Enter the END command to return to the Foreground Selection Panel and select another processor.
- Enter the RETURN command to go to the ISPF/PDF Primary Option Menu.
- Use the jump function (=) to choose any primary option.

Expanding Packed Data

Packed data is data in which PDF has replaced any repeating characters with a sequence showing how many times the character is repeated. Packing data lets you use direct access storage devices (DASDs) more efficiently because the stored data occupies less space than it would otherwise.

If the source data that you want to process is packed, it *must* be expanded before it can be successfully processed by any of the language processors. The expansion method you should use depends on whether your source data is:

- A sequential data set that contains expansion triggers

An *expansion trigger* is a keyword that tells PDF to expand additional data before copying, including, or imbedding it in the source data. Examples are INCLUDE and COPY statements, and SCRIPT .IM (imbed) control words. For information about defining your own expansion triggers, see *Installation and Customization*.

PDF does *not* recognize expansion triggers in data stored as a *sequential* data set. Therefore, for this type of data, you should follow these steps:

1. Manually expand the data that is to be copied, included, or imbedded in your source data. To do this, edit the source data, enter the PACK OFF command, and then save the data. When you have finished processing the data, you can repack it by editing it again and entering PACK ON.
 2. Enter YES in the SOURCE DATA PACKED field before invoking one of the language processors.
- Either of the following:
 - A sequential data set that does not contain expansion triggers
 - Any member of a partitioned data set, either with or without expansion triggers.

PDF *does* recognize expansion triggers in data stored as members of a *partitioned* data set. Also, if your source data does not contain expansion triggers, you do not have to be concerned with them. Therefore, for these two types of data, just enter YES in the SOURCE DATA PACKED field before invoking one of the language processors.

In each of the preceding situations, entering YES in the SOURCE DATA PACKED field causes PDF to expand packed source data before it is processed. For partitioned data sets, any included members are also expanded inline where the INCLUDE or COPY statements, .IM SCRIPT control words, or other user-defined trigger statements are found.

NW: 15457 DocId: 70002077 Page 599

Member expansion uses simple language scanners to find expansion triggers. If you specify that the source data is not packed, the ISRSCAN

Expanding Packed Data

program is used. However, if you specify that the source data is packed, member expansion uses the ISRLEMX program. These scanners do not have all the sophistication of the actual language processors. Therefore, unusual code or code that does not compile cannot be successfully processed by member expansion. Examples are trigger statements:

- With comments that extend onto the next line
- That have compiler instructions to change the content of the code to be included.

Compiler control statements and/or symbolic substitution are not considered during member expansion.

During member expansion, ISRLEMX creates a temporary data set to be used as input to the language processor. All members to be processed, including members imbedded with COPY or INCLUDE statements, are copied into this data set, expanded, and passed on to the language processor.

The temporary data set will have the same block size as the input data set that contains your source data. Therefore, you should make sure the block size of your input data set is smaller than or equal to the maximum block size supported by the direct access storage device (DASD) that will contain the temporary data set. Otherwise, processing can end abnormally (ABEND).

When using languages that allow multiple compilations, such as VS FORTRAN, you must put the program source ending statement in your original, or top-level, program. This statement cannot be in an included member.

The following table shows the languages processed by member expansion, their expansion triggers, syntax, and the input columns processed for fixed-record data and variable-record data.

Language	Expansion Trigger	Syntax	Input Columns Processed for F/FB Format	Input Columns Processed for V/VB Format
Assembler	COPY	COPY name	1 - 80	N/A
PL/I	%INCLUDE	%INCLUDE DDNAME(name); %INCLUDE name;	2 - 72	10 - 100
COBOL	COPY	COPY name.	7 - 72	N/A
VS FORTRAN	INCLUDE	INCLUDE (name)	1 - 72	N/A
Pascal	%INCLUDE	%INCLUDE name; %INCLUDE DDNAME(name);	1 - 72	1 - 100

Figure 97 (Part 1 of 2). Expansion Triggers and Syntax

Language	Expansion Trigger	Syntax	Input Columns Processed for F/FB Format	Input Columns Processed for V/VB Format
SCRIPT	.IM	.IM name .IM (name) .IM ('name')	1-reclength or 9-reclength	1-reclength or 1-(reclength-8)
All languages	User-trigger	User-trigger name	N/A	N/A

Figure 97 (Part 2 of 2). Expansion Triggers and Syntax

Restrictions on Member Expansion and Member Parts Lists

The following restrictions apply only to the member expansion and member parts listing functions:

- Restrictions that apply to all languages are:
 - Expansion triggers must follow their respective language coding conventions unless otherwise noted.
 - Multiple names and preprocessor variables on trigger statements are *not* permitted.
 - User triggers and their start column are specified at installation time and must be:
 - No more than 20 characters long
 - Uppercase with no imbedded blanks.

No part of the user trigger can be in a comment or continuation field.

 - Macros cannot be in packed form.
 - The trigger statement must be the only statement in the logical record (card image). No continuation is allowed into or from a trigger statement. Also, the trigger keyword must be the first character on the trigger statement that is not a blank and can be followed by only one statement delimiter.
 - For compilers that allow names longer than eight characters, the name is truncated at eight.
 - For compilers that allow uppercase and lowercase names, all referenced names are converted to uppercase.
- This restriction applies to Assembler only:
 - The user trigger cannot start in column 1.

Expanding Packed Data

- This restriction applies to FORTRAN only:
 - The member expansion function allows only the fixed form of coding.
- This restriction applies to PL/I, Pascal, and COBOL:
 - Free form coding is allowed except in trigger statements.
- Other COBOL restrictions are:
 - The name is truncated at eight characters or the first hyphen (-), whichever comes first.
 - The first statement in the COBOL program must be either an expansion trigger, a valid COBOL division header, a TITLE, a PROCESS, or a CBL statement. The expansion trigger can precede all other statements, but it must start in FIELD B.
 - In the COPY statement, the text-name is the only value processed. The statement must end on the same line as the COPY keyword with a period followed by a space. If any option is found, the COPY statement is not expanded.
 - In the IDENTIFICATION DIVISION, the division header or paragraph header statements must be blank except for the division or paragraph name. The trigger statement must be on the next line that is neither blank nor a comment.
 - In all other divisions, the trigger statement (line) can be on any line in the division.
 - If the WITH DEBUGGING MODE clause is not found in the SOURCE COMPUTER paragraph, all debug lines are passed to the compiler without being scanned for expansion triggers, as if they were comment lines. If the clause is found, valid trigger statements found on debug lines are expanded and a “D” is inserted in column 7 of all the non-comment, non-continuation lines included.
 - Any character found in FIELD A that is not a blank causes the end of the paragraph form of the NOTE statement.
- SCRIPT/VS restrictions are:
 - The “.im” statement must be the only statement in the logical record and must start in the first valid column. The first logical record is tested for line numbers, as follows:
 - For fixed-length records, if the last eight characters are all numeric, they are skipped for the complete library.
 - For variable-length records, if the first eight characters are all numeric, processing begins with column 9.

NW: 15457 DocId: 70002077 Page 602 The statements can be in either uppercase, lowercase, or mixed case.

Member Expansion Return Codes

Return Code	Explanation
0	Normal completion.
<i>n</i>	Where <i>n</i> = 1 - 15. Parm <i>n</i> was too long.
16	Too many parameters.
17	Too few parameters.
20	Severe error in expand module. An error message should be printed in the ISRLMSG data set.

Trigger Statement Errors

Some of the more common errors that occur are:

- Restricted option
- Statement on more than one line
- Referenced member name not found.

If an error occurs, the trigger statement is not expanded and is passed to the language processor.

In SCRIPT/VS, if the error was found in a user trigger, one blank line is inserted before and after the statement in question.

Input Data Sets

Input to a foreground processor is one of the following:

- A member of an ISPF library or "other" partitioned data set. If you do not specify a member name, PDF displays a member list.
- A sequential data set.

If an ISPF library is the input source, the member can be in any library in the concatenation sequence. You can include additional input by using:

- The COPY statement for Assembler and COBOL
- The INCLUDE statement for PL/I, FORTRAN, and Pascal
- The SCRIPT/VS imbed control word (.im)
- Macros
- Additional input libraries.

List Data Sets

No matter what the input source, you can always specify additional input libraries. They must be partitioned data sets that are *not* password protected. Be sure to specify the fully qualified data set name(s), enclosed in apostrophes, such as:

```
ADDITIONAL INPUT LIBRARIES:  
    ===>
```

For example, in Figure 101 on page 220, a concatenation sequence of three ISPF data sets and one additional input library has been specified. The concatenation order is:

```
ISPFDEMO.XXX.ASM  
ISPFDEMO.A.ASM  
ISPFDEMO.MASTER.ASM  
ISPFTEST.FLAG.ASM
```

The last data set in the concatenation sequence, ISPFTEST.FLAG.ASM, is entered as an additional input library at the bottom of the panel. Additional input libraries are always last in the sequence.

Before invoking a foreground processor, PDF scans the concatenated sequence of libraries to find the member to be processed. For this example, the member name is TOP. If member TOP first appears in data set ISPFDEMO.A.ASM, the following prompter command would be generated by Assembler XF. concatenation sequence

```
ASM  'ISPFDEMO.A.ASM(TOP) '  
      LIB('SYS1.MACLIB',  
          'ISPFDEMO.XXX.ASM',  
          'ISPFDEMO.A.ASM',  
          'ISPFDEMO.MASTER.ASM',  
          'ISPFTEST.FLAG.ASM')  
      LOAD('ISPFDEMO.XXX.OBJ(TOP)')  
      PRINT(LISTASM)  
      LIST,TEST,TERM,RENT
```

The processor options are passed to the prompter exactly as you specify them.

Note: The macro library SYS1.MACLIB is included in the concatenation sequence for Assembler only. When included, as the preceding prompter command example shows, it is always first in the sequence because of its large block size.

List Data Sets

In the LIST ID field, you can enter the name you want PDF to use to identify the list data set that will contain the foreground processor output. This name is passed to the foreground processor by either the LIST or PRINT option. The following rules apply:

- If the input data set is *partitioned* the LIST ID field is optional:
 - Leave the LIST ID field blank if you want PDF to use the input member name to identify the output list data set.

- Enter a LIST ID if you want to use a name other than the input member name to identify the output list data set.
- If the input data set is *sequential*, you must enter a LIST ID.

For best results, if you plan to debug your program later using COBOL interactive debug:

- Enter the name of the member being compiled in the LIST ID field if the input data set is partitioned.
- If the input data set is sequential, enter the name of the sequential data set.

Then, when you debug your program, use these same names in the PROG ID fields on the COBOL Interactive Debug panel.

ISPF names the listing:

`prefix.userid.listid.LIST`

where “prefix” is your TSO prefix, if you have one and if it is different from your user ID, “userid” is your user ID, and “listid” is the member name or the value in the LIST ID field.

If you are using the same list data set for multiple job steps, be aware that the DCB information can differ between the language processors and the linkage editor, causing an I/O error when trying to read the list data set. We suggest that you use a different list ID for each job step.

Password Protection

Input, object, interpretable text (ITEXT), and symbolic debug data sets can be password-protected. You can specify the password in the PASSWORD field on the foreground processor data entry panel. The password does not appear on the screen when you enter it, but PDF remembers it.

Since foreground processor panels have only one PASSWORD field, PDF prompts you if all data sets do not have the same password.

Object Data Sets

The following information about object data sets applies to all foreground assemblers and compilers. However, if you are using the VS FORTRAN or PL/I checkout compiler, you must enter OBJECT in the OBJECT field to generate an output object module. The two Assemblers and the other compilers generate object modules automatically.

Note: When used with the PL/I checkout compiler, the OBJECT option also generates an interpretable text (ITEXT) data set. See “Interpretable Text Data Sets” on page 227 if you need more information.

Foreground - TSO/E Information Center Facility

If you specify an ISPF library as the input source, PDF writes object output from the foreground assembler or compiler to a partitioned data set. This data set has the same name as the first library in the concatenation sequence, but has a type of OBJ. For example, if you specify PROJECT.LIB1.ASM as the first library name, the object output is placed in data set PROJECT.LIB1.OBJ. The member name of the object module is the same as the input member.

If you specify an “other” data set, the object output is placed in a data set of the same name, but with the last qualifier replaced by OBJ. If the data set name has only one qualifier, OBJ is appended as the last qualifier. For example, if you specify an input data set named OTHER.ASM or OTHER, the object output is placed in a data set named OTHER.OBJ. For partitioned data sets, the object output is stored in a member with the same name as the input member. For sequential data sets, the object output is stored in a sequential data set.

Foreground - TSO/E Information Center Facility

If the TSO/E Information Center Facility is installed, your installation can optionally allow PDF to substitute the panel shown in Figure 98 for the panel shown in Figure 96 on page 210. This panel is valid for all foreground processors except SCRIPT/VS and member parts list. See “Using the TSO/E Information Center Facility” on page 153 for information about the fields on this panel.

```
----- FOREGROUND PRINT OPTIONS -----
OPTION  ----> PD_

    PK - Print data set and keep          K - Keep data set (without printing)
    PD - Print data set and delete        D - Delete data set (without printing)

    If END command is entered, data set is kept without printing.

DATASET NAME: KRAMER.SPF101.LIST

PRINTER LOCATION  ---->
PRINTER FORMAT    ---->
NUMBER OF COPIES  ---->
```

Figure 98. Foreground Print Options Panel With TSO/E Information Center Facility

Assembler H (Option 4.1)

Foreground Assembler H is invoked from the Foreground Assembler H panel, shown in Figure 99. Assembler H does not use a prompter. For information about Assembler H allocation data sets, see Appendix D, “Allocation Data Sets” on page 367.

```

----- FOREGROUND ASSEMBLER H -----
COMMAND ---->

ISPF LIBRARY:
PROJECT ----> ISPFDEMO
GROUP ----> XXX      ----> A      ----> MASTER  ---->
TYPE ----> ASM
MEMBER ----> TOP      (Blank or pattern for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ---->

LIST ID ----> LISTASM_      PASSWORD ---->

ASSEMBLER OPTIONS: (Options OBJECT and LIST generated automatically)
----> TEST,TERM,RENT

ADDITIONAL INPUT LIBRARIES:
----> 'ISPFTEST.FLAG.ASM'
---->
    
```

Figure 99. Foreground Assembler H Panel

All the fields on this panel are explained in Chapter 2, “Libraries and Data Sets,” except LIST ID, which is explained in “List Data Sets” on page 216 , ADDITIONAL INPUT LIBRARIES, which is explained in “Input Data Sets” on page 215, and the following:

ASSEMBLER OPTIONS

Be careful *not* to enter the OBJECT and LIST options in this field. ISPF generates these options automatically. OBJECT writes the output object module to a partitioned data set. LIST writes the output listing to a list data set. See “Object Data Sets” on page 217 and “List Data Sets” on page 216 if you need more information. The available Foreground Assembler H options are:

ALIGN	DECK	FLAG	RENT	SYSPARM	TEST
BATCH	ESD	LINECOUNT	RLD	TERM	XREF

Figure 100. Foreground Assembler H Options

Foreground - Assembler XF (Option 4.1A)

Assembler XF (Option 4.1A)

Foreground Assembler XF is invoked from the Foreground Assembler XF panel, shown in Figure 101.

```
----- FOREGROUND ASSEMBLER XF -----
COMMAND ---->

ISPF LIBRARY:
PROJECT ----> ISPFDEMO
GROUP ----> XXX      ----> A      ----> MASTER      ---->
TYPE ----> ASM
MEMBER ----> TOP      (Blank or pattern for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ---->

LIST ID ----> LISTASM_      PASSWORD ---->

ASSEMBLER OPTIONS: (Options LIB, LOAD, and PRINT generated automatically)
----> LIST,TEST,TERM,RENT

ADDITIONAL INPUT LIBRARIES:
----> 'ISPFTEST.FLAG.ASM'
---->
```

Figure 101. Foreground Assembler XF Panel

All the fields on this panel are explained in Chapter 2, “Libraries and Data Sets,” except LIST ID, which is explained in “List Data Sets” on page 216 , ADDITIONAL INPUT LIBRARIES, which is explained in “Input Data Sets” on page 215, and the following:

ASSEMBLER OPTIONS

Be careful *not* to enter the LIB, LOAD, and PRINT options in this field. ISPF generates these options automatically. LIB specifies the input data set concatenation sequence. LOAD writes the output object module to a partitioned data set. PRINT writes the output listing to a list data set. See “Input Data Sets” on page 215, “Object Data Sets” on page 217, and “List Data Sets” on page 216 if you need more information. The available Foreground Assembler XF options are:

ALIGN	FLAG	LIST	MLOGIC	RLD	TERM
ALOGIC	LIBMAC	MACLIB	NUMBER	STMT	TEXT
BUFSIZE	LINECOUNT	MCALL	RENT	SYSPARM	XREF
ESD					

Figure 102. Foreground Assembler XF Options

VS COBOL II Compiler (Option 4.2)

ISPF generates a COBOL command from the values you enter on the Foreground VS COBOL II Compile panel, shown in Figure 103. VS COBOL II does not use a prompter. For information about VS COBOL II allocation data sets, see Appendix D, “Allocation Data Sets” on page 367.

```
----- FOREGROUND VS COBOL II COMPILE -----
COMMAND --->

ISPF LIBRARY:
PROJECT ---> ISPFDEMO
GROUP ---> XXX ---> A ---> MASTER --->
TYPE ---> COBOL
MEMBER ---> TOP (Blank or pattern for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME --->

LIST ID ---> LSTCOBOL_ PASSWORD --->

COMPILER OPTIONS: (Options LIB and OBJECT generated automatically)
TEST ---> TEST (TEST or NOTEST)
OTHER ---> RESIDENT

ADDITIONAL INPUT LIBRARIES:
----> 'ISPFTEST.FLAG.COBOL'
---->
```

Figure 103. Foreground VS COBOL II Compile Panel

All the fields on this panel are explained in Chapter 2, “Libraries and Data Sets,” except LIST ID, which is explained in “List Data Sets” on page 216 , ADDITIONAL INPUT LIBRARIES, which is explained in “Input Data Sets” on page 215, and the following:

TEST If you plan to run VS COBOL II interactive debug after you compile your program, enter TEST in the TEST field. Otherwise, enter NOTEST.

OTHER If you plan to run VS COBOL II interactive debug after you compile your program, enter RESIDENT in the OTHER field. Otherwise, just enter any other options you need.

Be careful *not* to enter the LIB and OBJECT options in the OTHER field. ISPF generates these options automatically. LIB specifies the input data set concatenation sequence. OBJECT writes the output object module to a partitioned data set. See “Input Data Sets” on page 215 and “Object Data Sets” on page 217 if you need more information.

Foreground - VS COBOL II Compiler (Option 4.2)

The available Foreground VS COBOL II compiler options are:

ADV	DUMP	MAP	RENT	TERMINAL
APOST	DYNAM	NUMBER	RESIDENT	TRUNC
BUF	FASTSRT	OFFSET	SEQUENCE	VBREF
COMPILE	FDUMP	OPTIMIZE	SIZE	WORD
DATA	FLAG	OUTDD	SOURCE	XREF
DECK	GRAPHIC	PFDSGN	SPACE	ZWB
DISK	LINECOUNT	QUOTE	SSRANGE	

Figure 104. Foreground VS COBOL II Compiler Options

OS/VS COBOL Compiler (Option 4.2A)

The Foreground OS/VS COBOL Compile panel is shown in Figure 105.

```
----- FOREGROUND OS/VS COBOL COMPILE -----
COMMAND ==>

ISPF LIBRARY:
PROJECT ==> ISPFDEMO
GROUP   ==> XXX      ==> A          ==> MASTER   ==>
TYPE    ==> COBOL
MEMBER  ==> TOP      (Blank or pattern for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ==>

LIST ID ==> LSTCOBOL_      PASSWORD ==>

COMPILER OPTIONS: (Options LIB, LOAD, and PRINT generated automatically)
TEST ==> TEST (TEST or NOTEST)
OTHER ==> SOURCE,XREF,LINECNT(73)

ADDITIONAL INPUT LIBRARIES:
==> 'ISPFTEST.FLAG.COBOL'
==>
```

Figure 105. Foreground OS/VS COBOL Compile Panel

All the fields on this panel are explained in Chapter 2, “Libraries and Data Sets,” except LIST ID, which is explained in “List Data Sets” on page 216, ADDITIONAL INPUT LIBRARIES, which is explained in “Input Data Sets” on page 215, and the following:

TEST If you plan to run COBOL interactive debug after you compile your program, enter TEST in the TEST field. Otherwise, enter NOTEST.

Note: Before you can specify TEST, you must have already used the Data Set utility (option 3.2) to allocate space for a symbolic debug data set. See “Symbolic Debug Data Sets” on page 242 if you need more information.

OTHER Enter any other options you need in the OTHER field, *except* LIB, LOAD, and PRINT. ISPF generates these options automatically. LIB specifies the input data set concatenation sequence. LOAD writes the output object module to a partitioned data set. PRINT writes the output listing to a list data set. See “Input Data Sets” on page 215, “Object Data Sets” on page 217, and “List Data Sets” on page 216 if you need more information.

Foreground - OS/VS COBOL Compiler (Option 4.2A)

The available Foreground OS/VS COBOL compiler options are:

ADV	DUMP	LCOL2	NUM	SOURCE	SYNTAX
APOST	DYNAM	LINECNT	OBJECT	SPACE1	SYS
BATCH	ENDJOB	LSTONLY	OPTIMIZE	SPACE2	TERM
BUF	FDECK	LSTCOMP	PMAP	SPACE3	TRUNC
CDECK	FLAGE	LVL	QUOTE	STATE	VBREF
CLIST	FLAGW	L120	RESIDENT	SUPMAP	VBSUM
COUNT	FLOW	L132	SEQ	SXREF	VERB
CSYNTAX	LCOL1	NAME	SIZE	SYMDMP	XREF
DMAP					

Figure 106. Foreground OS/VS COBOL Compiler Options

VS FORTRAN Compiler (Option 4.3)

The Foreground VS FORTRAN Compile panel is shown in Figure 107.

```

----- FOREGROUND VS FORTRAN COMPILE -----
COMMAND ---->

ISPF LIBRARY:
PROJECT ----> ISPFDEMO
GROUP  ----> XXX      ----> A      ----> MASTER  ---->
TYPE   ----> FORT
MEMBER ----> TOP      (Blank or pattern for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ---->

LIST ID ----> LISTFORT_      PASSWORD ---->

COMPILER OPTIONS:
OBJECT ----> OBJECT  (OBJECT or NOOBJECT)
OTHER  ----> SOURCE,TEST

ADDITIONAL INPUT LIBRARIES:
----> 'ISPFTEST.FLAG.FORT'
---->
    
```

Figure 107. Foreground VS FORTRAN Compile Panel

All the fields on this panel are explained in Chapter 2, “Libraries and Data Sets,” except LIST ID, which is explained in “List Data Sets” on page 216 , ADDITIONAL INPUT LIBRARIES, which is explained in “Input Data Sets” on page 215, and the following:

OBJECT PDF does not automatically generate any options for VS FORTRAN. Instead of generating an object module automatically, the VS FORTRAN compiler lets you decide whether to generate one. To generate an object module, enter OBJECT in the OBJECT field. To avoid generating an object module, enter NOOBJECT. See “Object Data Sets” on page 217 if you need more information.

OTHER If you plan to run FORTRAN interactive debug after you compile your program, enter TEST in the OTHER field, along with any other options you need. The available Foreground VS FORTRAN compiler options are:

DECK	FREE	LINECOUNT	NAME	SOURCE	TEST
FIPS	GOSTMT	LIST	OPTIMIZE	TERMINAL	XREF
FLAG	LANGVL	MAP			

Figure 108. Foreground VS FORTRAN Compiler Options

Foreground - PL/I Checkout Compiler (Option 4.4)

PL/I Checkout Compiler (Option 4.4)

The Foreground PL/I Checkout Compile panel is shown in Figure 109.

```
----- FOREGROUND PL/I CHECKOUT COMPILE -----
COMMAND ---->

ISPF LIBRARY:
PROJECT ----> ISPFDEMO
GROUP  ----> XXX      ----> A      ----> MASTER  ---->
TYPE   ----> PLIT
MEMBER ----> TOP      (Blank or pattern for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ---->

LIST ID ----> LISTPLIT_      PASSWORD ---->

EXECUTION PARMS:
---->

COMPILER OPTIONS:      (Options LIB and PRINT generated automatically)
OBJECT ----> OBJECT      (OBJECT or NOOBJECT)
OTHER  ----> MACRO,SOURCE,XREF

ADDITIONAL INPUT LIBRARIES:
----> 'ISPFTEST.FLAG.PLIT'
---->
```

Figure 109. Foreground PL/I Checkout Compile Panel

All the fields on this panel are explained in Chapter 2, “Libraries and Data Sets,” except LIST ID, which is explained in “List Data Sets” on page 216, ADDITIONAL INPUT LIBRARIES, which is explained in “Input Data Sets” on page 215, and the following:

EXECUTION PARMS

Enter any valid parameters that you want to pass to the program that you will compile.

OBJECT

The PL/I checkout compiler uses the OBJECT field instead of automatically generating the LOAD option. If you enter OBJECT in this field, ISPF generates an object module and an interpretable text (ITEXT) data set. To avoid generating these data sets, enter NOOBJECT. See “Object Data Sets” on page 217 and “Interpretable Text Data Sets” on page 227 if you need more information.

OTHER

Enter any other options you need in the OTHER field. However, be sure *not* to enter the LIB or PRINT options in this field. ISPF generates these options automatically. LIB specifies the input data set concatenation sequence. PRINT writes the output listing to a list data set. See “Input Data Sets” on page 215 and “List Data Sets” on page 216 if you need more information.

Foreground - PL/I Checkout Compiler (Option 4.4)

The available Foreground PL/I checkout compiler options are:

AGGREGATE	COUNT	INSOURCE	NEST	STEP
ASIS	DIAGNOSE	ISASIZE	NUMBER	STEPLINES
ATTRIBUTES	DUMP	LINECOUNT	OPTIONS	STMT
BLOCK	ERRORS	LMESSAGE	RUN	STORAGE
CAPS	ESD	MACRO	SEQUENCE	SYNTAX
CHARSET	FLAG	MARGINI	SIZE	TERMINAL
COMPATIBLE	FLOW	MARGINS	SMAN	VERIFY
COMPILE	FORMAT	MDECK	SMESSAGE	XREF
CONTROL	HALT	NAME	SOURCE	

Figure 110. Foreground PL/I Checkout Compiler Options

Interpretable Text Data Sets

When using the PL/I checkout compiler, you must enter **OBJECT** in the **OBJECT** field to generate an interpretable text (ITEXT) data set.

*Note: The **OBJECT** option also generates an output object module. See “Object Data Sets” on page 217 if you need more information.*

If you specify an ISPF library as the input source, PDF writes ITEXT output from the PL/I checkout compiler to a partitioned data set. This data set has the same name as the first library in the concatenation sequence, but has a type of ITEXT. For example, if you specify **PROJECT.LIB1.PLI** as the first library name, the ITEXT output is placed in data set **PROJECT.LIB1.ITEXT**. The member name of the ITEXT data set is the same as the input member.

If you specify an “other” data set, the ITEXT output is placed in a data set of the same name, but with the last qualifier replaced by ITEXT. If the data set name has only one qualifier, ITEXT is appended as the last qualifier. For example, if you specify an input data set named **OTHER.PLI** or **OTHER**, the ITEXT output is placed in a data set named **OTHER.ITEXT**. For partitioned data sets, the ITEXT output is stored in a member with the same name as the input member. For sequential data sets, the ITEXT output is stored in a sequential data set.

PL/I Optimizing Compiler (Option 4.5)

The Foreground PL/I Optimizing Compile panel is shown in Figure 111.

```
----- FOREGROUND PL/I OPTIMIZING COMPILE -----
COMMAND ---->

ISPF LIBRARY:
  PROJECT ----> ISPFDEMO
  GROUP   ----> XXX      ----> A      ----> MASTER   ---->
  TYPE    ----> PLIT
  MEMBER  ----> TOP      (Blank or pattern for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
  DATA SET NAME ---->

LIST ID ----> LISTPLIT_      PASSWORD ---->

COMPILER OPTIONS: (Options LIB, OBJECT, and PRINT generated automatically)
  OTHER ----> MACRO,SOURCE,XREF

ADDITIONAL INPUT LIBRARIES:
  ----> 'ISPFTEST.FLAG.PLIT'
  ---->
```

Figure 111. Foreground PL/I Optimizing Compile Panel

All the fields on this panel are explained in Chapter 2, “Libraries and Data Sets,” except LIST ID, which is explained in “List Data Sets” on page 216 , ADDITIONAL INPUT LIBRARIES, which is explained in “Input Data Sets” on page 215, and the following:

OTHER Enter any options you need in the OTHER field, *except* LIB, OBJECT, or PRINT. ISPF generates these options automatically. LIB specifies the input data set concatenation sequence. OBJECT writes the output object module to a partitioned data set. PRINT writes the output listing to a list data set. See “Input Data Sets” on page 215, “Object Data Sets” on page 217, and “List Data Sets” on page 216 if you need more information. The available Foreground PL/I optimizing compiler options are:

AGGREGATE	FLAG	LINECOUNT	NAME	SEQUENCE
ATTRIBUTES	FLOW	LIST	NEST	SIZE
CHARSET	GONUMBER	LMESSAGE	NOSPIE	SOURCE
COMPILE	GOSTMT	MACRO	NOSTAE	STMT
CONTROL	GRAPHIC	MAP	NUMBER	STORAGE
COUNT	IMPRECISE	MARGINI	OFFSET	SYNTAX
DECK	INCLUDE	MARGINS	OPTIMIZE	TERMINAL
DUMP	INSOURCE	MDECK	OPTIONS	XREF
ESD	INTERRUPT			

Figure 112. Foreground PL/I Optimizing Compiler Options

Pascal/VS Compiler (Option 4.6)

The Foreground Pascal/VS Compile panel is shown in Figure 113.

```

----- FOREGROUND PASCAL/VS COMPILE -----
COMMAND ---->

ISPF LIBRARY:
PROJECT ----> ISPFDEMO
GROUP ----> XXX      ----> A      ----> MASTER  ---->
TYPE ----> PAS
MEMBER ----> TOP      (Blank or pattern for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ---->

LIST ID ----> LISTPAS_      PASSWORD ---->

COMPILER OPTIONS: (Options LIB, OBJECT, and PRINT generated automatically)
OTHER ----> SOURCE,XREF(SHORT)

ADDITIONAL INPUT LIBRARIES:
----> 'ISPFTEST.FLAG.PAS'
---->
  
```

Figure 113. Foreground Pascal/VS Compile Panel

All the fields on this panel are explained in Chapter 2, "Libraries and Data Sets," except LIST ID, which is explained in "List Data Sets" on page 216, ADDITIONAL INPUT LIBRARIES, which is explained in "Input Data Sets" on page 215, and the following:

OTHER Enter any options you need in the OTHER field, *except* LIB, OBJECT, or PRINT. ISPF generates these options automatically. LIB specifies the input data set concatenation sequence. OBJECT writes the output object module to a partitioned data set. PRINT writes the output listing to a list data set. See "Input Data Sets" on page 215, "Object Data Sets" on page 217, and "List Data Sets" on page 216 if you need more information. The available Foreground Pascal/VS compiler options are:

Default Options	Other Options
CHECK	NOCHECK
GOSTMT	NOGOSTMT
LANGLVL(EXTENDED)	LANGLVL(STANDARD)
LINECOUNT(60)	LINECOUNT(N)
MARGINS(1,72)	MARGINS(M,N)
NODEBUG	DEBUG
NOLIST	LIST
OPTIMIZE	NOOPTIMIZE
PAGEWIDTH(128)	PAGEWIDTH(N)
PXREF	NOPXREF
SEQUENCE(M,N)	NOSEQUENCE

Figure 114 (Part 1 of 2). Foreground Pascal/VS Compiler Options

Default Options

SOURCE
WARNING
XREF(SHORT)

Other Options

NOSOURCE
NOWARNING
XREF(LONG)/NOXREF

Figure 114 (Part 2 of 2). Foreground Pascal/VS Compiler Options

Linkage Editor (Option 4.7)

The Foreground Linkage Editor is invoked from the Foreground Linkage Edit panel, shown in (Figure 115).

```
----- FOREGROUND LINKAGE EDIT -----
COMMAND ---->

ISPF LIBRARY:
PROJECT ----> ISPFDEMO
GROUP ----> MYLIB ----> MASTER ---->
TYPE ----> OBJ
MEMBER ----> MAINPROG (Blank or pattern for member selection list)

OTHER PARTITIONED DATA SET:
DATA SET NAME ---->

LIST ID ----> LLLL_ PASSWORD ---->

LINKAGE EDITOR OPTIONS: (Options LOAD, LIB, and PRINT generated automatically)
----> MAP,LET,TEST,RENT,DCBS(3072)

ADDITIONAL INPUT LIBRARIES: (LOAD libraries only)
---->
---->
```

Figure 115. Foreground Linkage Edit Panel

All the fields on this panel are explained in Chapter 2, “Libraries and Data Sets,” except LIST ID, which is explained in “List Data Sets” on page 216, ADDITIONAL INPUT LIBRARIES, which is explained in “Input Data Sets” on page 215, and the following:

LINKAGE EDITOR OPTIONS

Enter any options you need, *except* LOAD, LIB, or PRINT. ISPF generates these options automatically. LOAD writes the output object module to a partitioned data set.

Note: Sequential data sets are invalid when using the Linkage Editor.

LIB specifies the input data set concatenation sequence. PRINT writes the output listing to a list data set. See “Object Data Sets” on page 217 and “List Data Sets” on page 216 if you need more information. The available Foreground Linkage Editor options are:

AC	LET	NE	PLIBASE	RENT	TERM
COBLIB	LIST	NONE	PLICMIX	REUS	TEST
DC	MAP	OL	PLILIB	SCTR	XCAL
DCBS	NCAL	OVLY	REFR	SIZE	XREF
FORTLIB					

Figure 116. Foreground Linkage Editor Options

Linkage Editor Concatenation Sequence

The concatenation sequence used by PDF to find the member for input to the Linkage Editor is:

```
project-name.lib1-name.type  
project-name.lib2-name.type  
(and so forth)
```

where “type” is whatever you specify on the panel. For example, it can be OBJ or some other type containing Linkage Editor language (LEL) control statements.

If the type is not OBJ, an “OBJECT” DDNAME is automatically allocated to ease the use of the following Linkage Editor control statements:

```
INCLUDE OBJECT(member-name)
```

For example:

```
PROJECT ===>  
GROUP   ===>           ===>           ===>           ===>  
TYPE    ===>  
MEMBER  ===>
```

In this example, PDF searches data sets XYZ.MYLIB.LEL and XYZ.MASTER.LEL to find member TOP, which should contain LEL control statements. Also, PDF allocates to DDNAME “OBJECT” (DISP=SHR) the following concatenated sequence of object libraries:

```
XYZ.MYLIB.OBJ  
XYZ.MASTER.OBJ
```

This concatenated sequence is searched *by the Linkage Editor* if member TOP contains INCLUDE OBJECT(member-name) statements.

The concatenation sequence passed to the Linkage Editor via the LIB parameter has a type qualifier of LOAD and includes the system libraries you specify, as follows:

```
LIB('project-name.lib1-name.LOAD',  
    'project-name.lib2-name.LOAD',  
  
    .  
    .  
    .  
    and so forth,  
  
    .  
    .  
    .  
    'syslib1-name',  
  
    .  
    .  
    .  
    and so forth)
```

This concatenation sequence is used by the Linkage Editor to resolve automatic call references.

SCRIPT/VS (Option 4.9)

Use of this facility requires the installation of the Document Composition Facility (DCF) program product and its component text processing program, SCRIPT/VS, with the Foreground Environment Feature.

When you select the SCRIPT/VS option, the first panel displayed is the SCRIPT/VS Processor panel (Figure 117).

```

----- SCRIPT/VS PROCESSOR -----
COMMAND ---->

STYLE ---->                                DISPLAY STYLE OPTIONS ----> NO   (Yes or No)

ISPF LIBRARY:
PROJECT ----> ISPFDEMO
GROUP ----> XXX      ----> TEST      ----> MASTER      ---->
TYPE ----> TEXT
MEMBER ----> TOP_      (Blank or pattern for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ---->

DATA SET PASSWORD ---->                    (If password protected)

LIST ID ---->                                (Blank defaults to member name)

BROWSE ----> YES                                (YES or NO)
  
```

Figure 117. SCRIPT/VS Processor Panel

All the fields on this panel are explained in Chapter 2, "Libraries and Data Sets," except LIST ID, which is explained in "List Data Sets" on page 216, and STYLE, DISPLAY STYLE OPTIONS, and BROWSE, which are explained in the following sections.

SCRIPT/VS Processing Sequence

A *style* contains options that tell SCRIPT/VS how to format a document for display or printing. These options include the use of fonts, white space, line lengths, and so forth.

The values you put in the STYLE, DISPLAY STYLE OPTIONS, and BROWSE fields determine the following SCRIPT/VS processing sequence:

1. For the STYLE field, you can do one of the following:
 - Enter the name of an existing style.

You can enter the name of a style you have created or one of the styles SCRIPT/VS creates for you: DRAFT and FINAL. These two styles correspond to the formatting options available in the previous release of SCRIPT/VS Foreground Processing. If you have not defined these options before or if this is your first release of ISPF/PDF, the default values for the SCRIPT/VS formatting options are set for you.

Foreground - SCRIPT/VS (Option 4.9)

If you enter the name of an existing style in the STYLE field, that style is used for formatting.

- Enter the name of a new style you want to define.

If you enter a new style name, the name is added to your style list. The new style uses SCRIPT/VS formatting options that are equal to the formatting options of the last style. Step 2 explains what to do to change these options.

- Leave the STYLE field blank.

If you leave the style field blank, PDF displays the Select SCRIPT/VS Formatting Style panel. This panel displays a list of the available styles. See “Selecting a Formatting Style” on page 237 if you need more information.

2. Enter either YES or NO in the DISPLAY STYLE OPTIONS field.

If you enter YES, PDF displays the SCRIPT/VS Options for Style panel. This panel shows the options that are currently being used and allows you to change them. See “Changing Style Options” on page 238 if you need more information.

If you enter NO, PDF does not display the SCRIPT/VS Options for Style panel.

3. Enter the appropriate ISPF library and concatenation sequence or data set name(s). You can display a member list by omitting the member name or by using a pattern. See Chapter 2, “Libraries and Data Sets” if you need help entering library or data set names, “Input Data Sets” on page 215 for more information about the concatenation sequence, and “Displaying Member Lists” on page 38 for more information about displaying member lists.
4. Enter your password in the PASSWORD field if your input data set is password-protected. See “Password Protection” on page 217 if you need more information.
5. Use the LIST ID field to tell PDF what to name the output SCRIPT/VS listing. See “List Data Sets” on page 216 if you need more information.
6. Enter either YES or NO in the BROWSE field.

If you enter YES, PDF displays your output in Browse mode after it has formatted.

If you enter NO, PDF skips Browse mode and displays a Foreground Print Options for Style panel, shown in Figure 118 on page 236.

7. Once all the input parameters have been specified, press the ENTER key to invoke SCRIPT/VS.

8. Communication with SCRIPT/VS is in line-I/O mode. Each time you see three asterisks, press the ENTER key. These asterisks, which usually

appear at the bottom of the screen, show that TSO is waiting for you to clear the screen before it can proceed.

If the Session Manager is installed and if you specified Session Manager mode on the Foreground Selection Panel, SCRIPT/VS and all PF and PA keys are under control of the Session Manager. When formatting is complete, you are prompted to enter a null line to return to PDF control.

If the Session Manager is not invoked, the PA and PF keys have their usual TSO-defined meanings; generally, the PF keys are treated the same as the ENTER key.

9. One or both of the following panels may appear, depending on your treatment of the STYLE and DISPLAY STYLE OPTIONS fields. If both appear, they will be in the following sequence:
 - a. Select SCRIPT/VS Formatting Style
 - b. SCRIPT/VS Options for Style

See "Selecting a Formatting Style" on page 237 and "Changing Style Options" on page 238 if you need information about using these panels. When you are finished with each panel, press the ENTER key.

10. If SCRIPT/VS generated an output listing and if you specified YES in the BROWSE field, the output is displayed automatically in Browse mode. Otherwise, continue with the next step.

Note: If SCRIPT/VS formatting ends abnormally, PDF displays a message in the upper-right corner of the screen and does not enter Browse mode. The list data set is retained, but the Foreground Print Options for Style panel (see step 11) is not displayed.

You can scroll the output up or down using the scroll commands. All the Browse commands are available to you. When you finish browsing the listing, enter the END command.

11. An optional print utility exit can be installed by your system programmer. If this exit is installed, it may cause SCRIPT/VS's response to differ from the following descriptions. See *Installation and Customization* for more information about the print utility exit.

Another factor that can affect SCRIPT/VS's performance is whether the TSO/E Information Center Facility is installed. If the TSO/E Information Center Facility is installed, your installation can optionally allow PDF to display a panel for submitting the TSO/E Information Center Facility information with the print request. See Figure 121 on page 239 for an example of this panel and "Using the TSO/E Information Center Facility" on page 153 for information about the fields on this panel.

If the TSO/E Information Center Facility is not installed, SCRIPT/VS displays the panel shown in Figure 118 on page 236.

Foreground - SCRIPT/VS (Option 4.9)

```
----- FOREGROUND PRINT OPTIONS FOR STYLE: DRAFT -----
COMMAND ----> _

PK - Print data set and keep          K - Keep data set (without printing)
PD - Print data set and delete         D - Delete data set (without printing)

If END command is entered, data set is kept without printing.

DATA SET NAME: ISPFDEMO.TOP.LIST

PRINT MODE ----> (BATCH or LOCAL)

SYSOUT CLASS ---->
PRINTER ID ----> (For 328x printer)

JOB STATEMENT INFORMATION: (Required for system printer)
----> //userida JOB (ACCOUNT),'NAME'
----> /**
----> /**
----> /**
```

Figure 118. Foreground Print Options for Style Panel

The Foreground Print Options for Style panel allows you to optionally print the formatted document and specify its disposition. On this panel, the DATA SET NAME field shows the name of the list data set that contains the SCRIPT/VS output. In the COMMAND field, enter one of the options shown at the top of the panel.

In the PRINT MODE field, enter either:

- **BATCH** to submit your print request as a background job.

If you choose BATCH, specify a valid SYSOUT CLASS and job statement information. Specifying BATCH causes SCRIPT/VS to ignore the PRINTER ID field.

SCRIPT/VS list data sets are formatted DCB=RECFM=VBM. Unless the line count is altered, the formatted page length may exceed the JES line count and cause duplicate page ejects. Therefore, specify the following job statement information to prevent JES line counting:

- **LOCAL** to print the output on a local printer.

If you choose LOCAL, specify the ID of a local printer. Specifying LOCAL causes SCRIPT/VS to ignore the SYSOUT CLASS field. Job statement information is ignored.

Page spacing will probably vary from the expected format because of differences between 328x printers and 1403 or 3800 printers used as a formatting guide.

See “Ending PDF or a PDF Function” on page 22 if you need information about the options at the top of the panel or the SYSOUT CLASS, PRINTER ID, and JOB STATEMENT INFORMATION fields.

When you press the ENTER key, the SCRIPT/VS Processor panel is displayed again. A message indicating completion of the process is displayed in the upper-right corner of the screen.

12. You can do one of the following:

- Enter other parameters and invoke SCRIPT/VS again.
- Enter the END command to return to the Foreground Selection Panel and select another processor.
- Enter the RETURN command to go to the ISPF/PDF Primary Option Menu.
- Use the jump function (=) to choose any primary option.

Selecting a Formatting Style

Use the Select SCRIPT/VS Formatting Style panel (Figure 119) to see which styles are available and to select or delete styles as necessary.

```
----- SELECT SCRIPT/VS FORMATTING STYLE ----- ROW 1 OF 2
COMMAND ===> _ SCROLL ===> PAGE

Valid Line Commands: S - Use this STYLE for formatting
                    D - Delete

Line
Cmd  Style      Description
*    DRAFT      Draft SCRIPT/VS document options
*    FINAL      Final SCRIPT/VS document options
***** BOTTOM OF DATA *****
```

Figure 119. Select SCRIPT/VS Formatting Style Panel

The Select SCRIPT/VS Formatting Style panel is a list that can be scrolled and contains all the styles available to you. Each style is a set of predefined formatting options.

Type either **S** or **D** in the Line Cmd field and press the ENTER key to select or delete a style, respectively. You can only select one style at a time for formatting. However, one or more styles can be deleted at the same time.

Foreground - SCRIPT/VS (Option 4.9)

The display fields on the Select SCRIPT/VS Formatting Style panel contain the following information:

Style	The names of styles that you can either select or delete.
Description	A reminder of the purpose of each style. Type over the description to change it.

Changing Style Options

Use the SCRIPT/VS Options for Style panel to:

- See which options are currently being used for the style you chose
- Change the options as needed.

Figure 120 shows the options available for the DRAFT style.

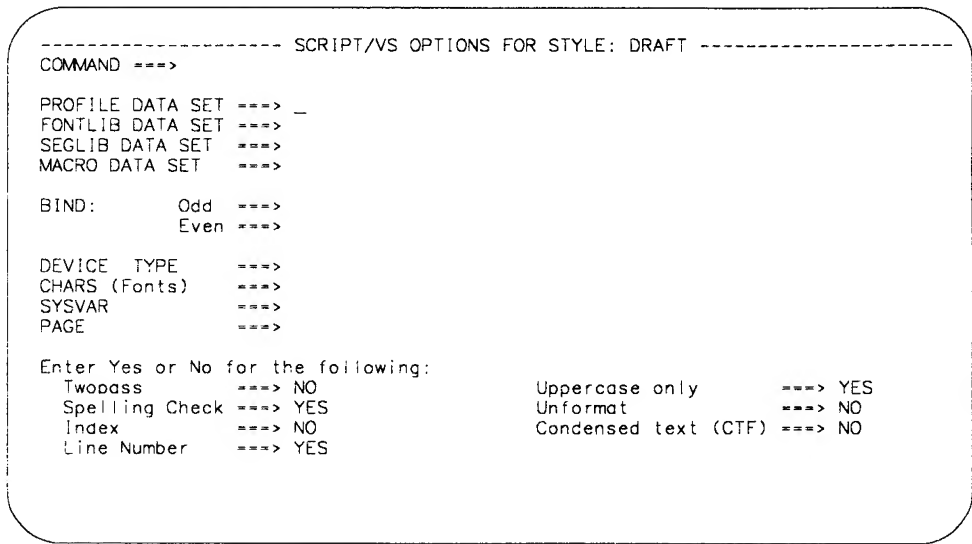


Figure 120. SCRIPT/VS Options for Style Panel: Draft

The fields on the SCRIPT/VS Options for Style panel represent SCRIPT/VS formatting options, all of which are optional. For a complete description of these options, see *Document Composition Facility: Generalized Markup Language Starter Set User's Guide*.

If you enter the END command from the SCRIPT/VS Options for Style panel, changes on this panel are not saved. If the style is new, it is saved with default formatting options.

If you press the ENTER key from the SCRIPT/VS Options for Style panel, SCRIPT/VS processes the data set, and then one of the following occurs:

- A Browse panel is displayed if you specified YES in the BROWSE field on the SCRIPT/VS Processor panel. When you finish browsing the SCRIPT/VS formatted output, a Foreground Print Options for Style panel is displayed.

- A Foreground Print Options for Style panel is displayed if you specified NO in the BROWSE field on the SCRIPT/VS Processor panel.

See step 11 on page 235 if you need more information about printing SCRIPT/VS output.

Using SCRIPT/VS with the TSO/E Information Center Facility

If the TSO/E Information Center Facility is installed, your installation can optionally allow PDF to substitute the panel shown in Figure 121 for the panel shown in Figure 118 on page 236. See “Using the TSO/E Information Center Facility” on page 153 for information about the fields on this panel.

```
----- FOREGROUND PRINT OPTIONS FOR STYLE: DRAFT -----
OPTION ----> _

  PK - Print data set and keep          K - Keep data set (without printing)
  PD - Print data set and delete        D - Delete data set (without printing)

  If END command is entered, data set is kept without printing.

DATA SET NAME: ISPFDEMO.TOP.LIST

PRINTER LOCATION      ---->
PRINTER FORMAT        ---->
NUMBER OF COPIES      ---->
```

Figure 121. Foreground Print Options for Style Panel With the TSO/E Information Center Facility

VS COBOL II Interactive Debug (Option 4.10)

To run VS COBOL II interactive debug in foreground, the VS COBOL II compiler, Release 2, must be both installed and accessible, because:

- You must compile your program by using the VS COBOL II compiler (option 4.2 or option 5.2) with the TEST and RESIDENT options before running VS COBOL II interactive debug. Debug output from the compilation is stored in the object module, which ISPF generates automatically.
- The VS COBOL II compiler contains the Debug Productivity Aid (DPA) facility, which PDF accesses when you run VS COBOL II interactive debug in the foreground. The panel shown in Figure 122 is displayed if PDF cannot access DPA.

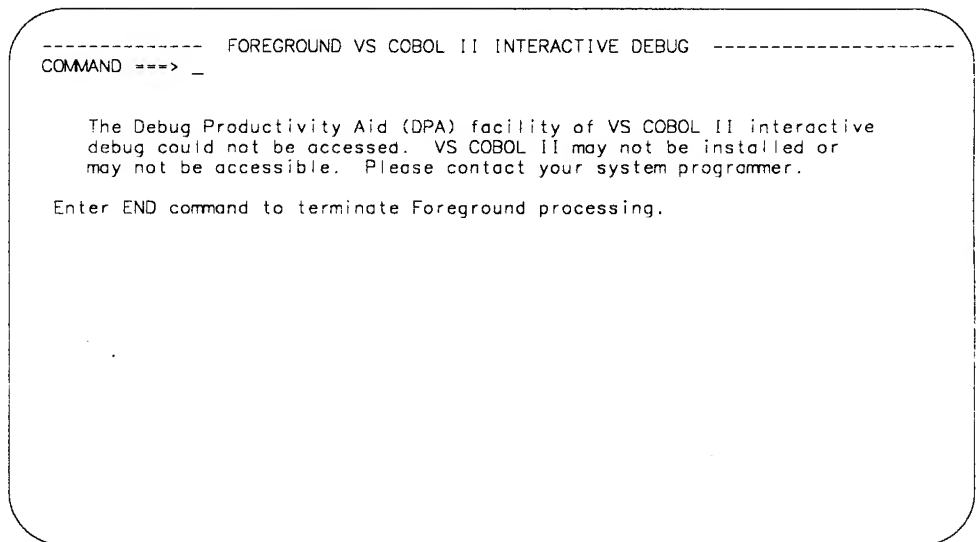


Figure 122. Foreground VS COBOL II Interactive Debug Panel

All VS COBOL II interactive debug processing in the foreground is under DPA's control. DPA displays a series of interactive panels. When processing is complete, return to step 14 on page 209.

COBOL Interactive Debug (Option 4.10A)

Before you can run COBOL interactive debug, you must first do the following, in the order shown:

1. Allocate a symbolic debug data set and, optionally, a print output data set by using the Data Set utility (option 3.2). See "Symbolic Debug Data Sets" on page 242 and "Print Output Data Sets" on page 243 if you need more information.
2. Compile the program by using the OS/VS COBOL compiler (option 4.2A or option 5.2A) with the TEST option.
3. Use the linkage editor (option 4.7 or option 5.7) to generate an output load module, which COBOL interactive debug will use as input.

The COBOL Interactive Debug panel is shown in Figure 123.

```
----- COBOL INTERACTIVE DEBUG -----
COMMAND ===>

ISPF LIBRARY:
  PROJECT ===> ISPFDEMO
  GROUP   ===> XXX           (TYPE = LOAD assumed)
  MEMBER  ===> TOP           (Blank or pattern for member selection list)

OTHER PARTITIONED DATA SET:
  DATA SET NAME ===>

PROG ID  ===> LSTCOBOL  ===>          ===>          ===>

PRINT ID ===>                                PASSWORD ===>

SOURCE   ===> YES_      (YES or NO)

EXECUTION PARMS:
  ===>

Notes: 1. 'PREFIX.PRINTID.TESTLIST' must exist if "PRINT ID" is specified.
       2. 'PREFIX.PROGID.LIST' must exist for each program specified if
          "YES" is specified in "SOURCE" field.
```

Figure 123. COBOL Interactive Debug Panel

All the fields on this panel are explained in Chapter 2, "Libraries and Data Sets," except PROG ID, PRINT ID, SOURCE, and EXECUTION PARMS, which are explained in the following sections.

COBOL Interactive Debug Processing Sequence

Fill in the fields on the COBOL Interactive Debug panel as follows:

1. Enter the ISPF library or data set name that contains the input load module generated by the linkage editor. You can display a member list by omitting the member name or by using a pattern. See Chapter 2, "Libraries and Data Sets" if you need help entering library or data set names, "Object Data Sets" on page 217 for more information about object modules, and "Displaying Member Lists" on page 38 for more information about displaying member lists.

2. The PROG ID field tells PDF the name(s) of the sequential list data set(s) generated by the OS/VS COBOL compiler. You can enter up to four PROG ID names if you compiled a partitioned data set member or a sequential data set for each name. See "List Data Sets" on page 216 if you need more information.
3. The PRINT ID field is optional. This field tells PDF the name of a sequential data set to which PDF writes the print output from the debug session. This data set must be preallocated. See "Print Output Data Sets" on page 243 if you need more information.
4. Enter your password in the PASSWORD field if your input data set is password-protected. See "Password Protection" on page 217 if you need more information.
5. The SOURCE field tells PDF whether to allocate the list data sets specified in the PROG ID field. If you enter YES, these data sets must already exist. Otherwise, enter NO.
6. The EXECUTION PARMS field is remembered from one session to another. Therefore, you do not need to change this field unless the parameters you need are not displayed. Enter any parameters that you want PDF to pass to the program being debugged.
7. To continue COBOL interactive debug, return to step 14 on page 209.

Symbolic Debug Data Sets

If you want to run COBOL interactive debug on a program compiled with the OS/VS COBOL compiler, you must use the Data Set utility (option 3.2) to allocate a symbolic debug data set *before* compiling the program. Then, when you compile the program, enter TEST in the TEST field on the Foreground OS/VS COBOL Compile panel. The TEST parameter generates the debug output, which PDF stores in the symbolic debug data set you allocated.

Note: You do not need to allocate a symbolic debug data set for programs compiled with VS COBOL II because the debug output, if requested, is stored in the OBJECT module, which ISPF generates automatically.

When you allocate the data set, specify the same name as the data set that contains your COBOL program, but:

- For an ISPF library, enter:
 - The GROUP name you will specify in the first GROUP field on the Foreground OS/VS COBOL Compile panel
 - SYM in the TYPE field.
- For an "other" partitioned or sequential data set, use SYM to replace the last qualifier. For example, if COBOL.INPUT or COBOL is the

input data set name, allocate COBOL.SYM as the symbolic debug data set.

Use the following values to allocate symbolic debug data sets:

```
RECORD FORMAT      ===>
RECORD LENGTH      ===>
BLOCK SIZE         ===>
```

For partitioned data sets, including ISPF libraries, the debug output is stored in a member with the same name as the input member. For sequential data sets, the debug output is stored in a sequential data set.

When you run COBOL interactive debug, the name(s) you put in the PROG ID field on the COBOL Interactive Debug panel must be the same as the input member name(s) if you are to create a correct SYM data set.

Print Output Data Sets

ISPF writes the print output from a debug session to a sequential data set, if you:

- Allocate the data set, using the Data Set utility (option 3.2), before you run COBOL interactive debug
- Enter, in the PRINT ID field on the COBOL Interactive Debug panel, the name of the data set you allocated.

You can avoid generating the print output by leaving the PRINT ID field blank, even if you allocated the data set.

The last qualifier in the name of the data set you allocate must be TESTLIST. For example, if you allocate a sequential data set named DEBUG1.TESTLIST and then specify the PRINT ID as:

```
PRINT ID ===> DEBUG1
```

ISPF writes the print output to a sequential data set named:

```
'prefix.userid.DEBUG1.TESTLIST'
```

where "prefix" is your TSO data set prefix, if you have one and if it is different from your user ID, and "userid" is your TSO user ID. Use the following values to allocate print output data sets:

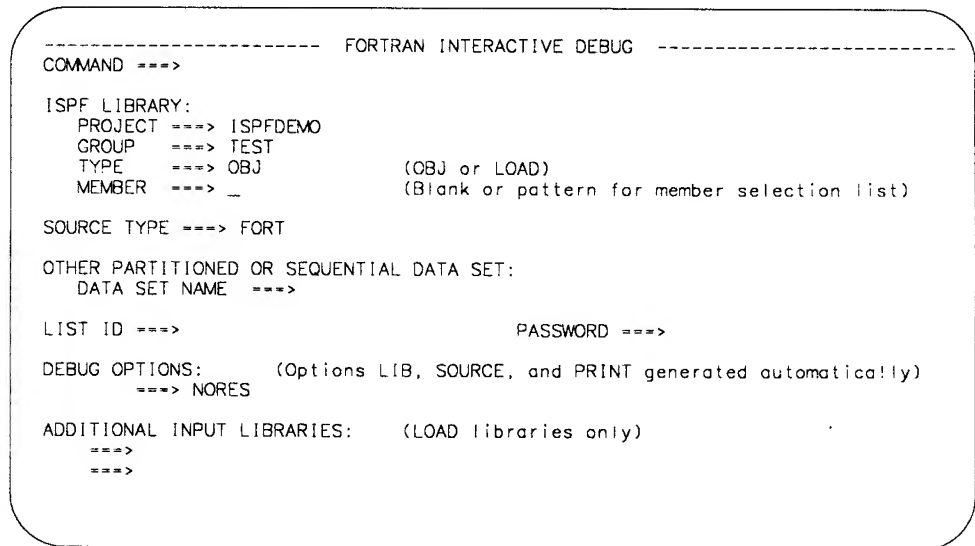
```
RECORD FORMAT      ===> 121
RECORD LENGTH      ===> 121
BLOCK SIZE         ===> 3146
```

The value you put in the BLOCK SIZE field should be a multiple of 121, the record length. Therefore, if your print output data is too large to fit within the recommended block size (3146), increase this amount by using a multiple of 121, such as 3267 or 3388.

FORTRAN Interactive Debug (Option 4.11)

Before you can run FORTRAN interactive debug, you must first compile the program using the VS FORTRAN compiler (option 4.3 or option 5.3) with the OBJECT and TEST options.

The FORTRAN interactive debug option supports both FORTRAN Interactive Debug Version 2 (5668-903) and FORTRAN Interactive Debug Version 1 (5734-F05). PDF looks for Version 2 first, then Version 1, and finally its own Debug Dialog, which displays the panel shown in Figure 124.



```
----- FORTRAN INTERACTIVE DEBUG -----
COMMAND ---->

ISPF LIBRARY:
  PROJECT ----> ISPFDEMO
  GROUP   ----> TEST
  TYPE    ----> OBJ          (OBJ or LOAD)
  MEMBER  ----> _           (Blank or pattern for member selection list)

SOURCE TYPE ----> FORT

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
  DATA SET NAME ---->

LIST ID ---->                PASSWORD ---->

DEBUG OPTIONS:      (Options LIB, SOURCE, and PRINT generated automatically)
  ----> NORES

ADDITIONAL INPUT LIBRARIES:  (LOAD libraries only)
  ---->
  ---->
```

Figure 124. FORTRAN Interactive Debug Panel

All the fields on this panel are explained in Chapter 2, “Libraries and Data Sets,” except LIST ID, which is explained in “List Data Sets” on page 216, ADDITIONAL INPUT LIBRARIES, which is explained in “Input Data Sets” on page 215, and the following:

DEBUG OPTIONS

Enter any options you need, *except* LIB, SOURCE, or PRINT. ISPF generates these options automatically. LIB specifies the input data set concatenation sequence. SOURCE specifies the input source program, whose type is identified in the SOURCE TYPE field. PRINT writes the output listing to a list data set. See “Input Data Sets” on page 215 and “List Data Sets” on page 216 if you need more information. The available Foreground FORTRAN interactive debug options are:

NORES RES

Figure 125. Foreground FORTRAN Interactive Debug Options

Member Parts List (Option 4.12)

The member parts list shows the following information for each source program module specified:

- The names of the modules it calls or includes
- The names of the modules that call or include it.

The languages permitted in the member expansion function also are permitted in the member parts list function, and the expansion triggers have the same restrictions. Besides the expansion triggers, the member parts list also uses the CALL statements in Assembler, PL/I, COBOL, and VS FORTRAN. The format of the CALL statement is:

CALL name

where the delimiter after the name can be either a left parenthesis, a blank, or a valid statement delimiter. In COBOL, the CALL statement is valid only in the PROCEDURE DIVISION, and the statements CALL PGMA and CALL 'PGMA' both result in a reference to the member name PGMA.

When you select the Foreground Member parts list option (4.12), the panel shown in Figure 126 is displayed.

```
----- FOREGROUND MEMBER PARTS LIST -----
OPTION ==> _

  1 - Browse/Print member parts
  2 - Write member parts data set

ISPF LIBRARY:
PROJECT  ==>
GROUP    ==>
TYPE     ==>
MEMBER   ==>
                                     (Blank for member selection list)

LANGUAGE  ==>
                                     (Defaults to TYPE value)

GROUPS FOR PRIMARY MEMBERS ==> 1
                                     (1, 2, 3, or 4)

OUTPUT DATA SET:
DATA SET NAME ==>
                                     (Option 2 only)
```

Figure 126. Foreground Member Parts List Panel

The member parts list does not use the SOURCE DATA PACKED field on the Foreground Selection Panel; both packed and unpacked data sets can be read.

Foreground - Member Parts List (Option 4.12)

- A parts list is requested for all members in the first data set.

You can scroll the output up or down using the scroll commands. All the Browse commands are available to you. When you finish browsing the listing, enter the END command and continue with step 17 on page 209.

- Option 2 (Write member parts data set) produces an intermediate sequential member parts list in the data set you named in step 5 on page 246. This data set can be either a sequential data set or a member of a partitioned data set.

If the data set has not been allocated, option 2 allocates it with a logical record length (LRECL) of 17, a block size (BLKSIZE) of 3009, and a record format (RECFM) of FB. The format of the records is shown in the following table:

Field Name	Format	Description
Member name	CHAR(8)	Subject member.
Called by or calls member name	CHAR(8)	Referenced member.
Call flag	BIT(1)	Found on a CALL statement.
Include flag	BIT(1)	Found by INCLUDE or COPY.
Not found flag	BIT(1)	Referenced member not found.
From flag	BIT(1)	Subject member called from referenced member.
To flag	BIT(1)	Referenced member called from subject member.
COBOL flag	BIT(1)	Member referenced outside valid COBOL division.
Reserved	BIT(2)	Field that is reserved.

Figure 128. Foreground Member Parts List Record Formats

- You can do one of the following:
 - Enter other parameters and invoke the same processor.
 - Enter the END command to return to the Foreground Selection Panel and select another processor.
 - Enter the RETURN command to go to the ISPF/PDF Primary Option Menu.
 - Use the jump function (=) to choose any primary option.

Foreground - Member Parts List (Option 4.12)

7. Communication with the Foreground Member Parts List processor is in line-I/O mode. Each time you see three asterisks, press the ENTER key. These asterisks, which usually appear at the bottom of the screen, show that TSO is waiting for you to clear the screen before it can proceed.
8. The option you chose in step 1 on page 246 determines what happens next:

Note: If the Foreground Member Parts List processing program ends abnormally, PDF displays a message in the upper-right corner of the screen and does not enter Browse mode. The list data set is retained, but the Foreground Print Options panel (see step 17 on page 209) is not displayed.

- a. Option 1 (Browse/print member parts) creates the member parts list and displays it in Browse mode. Figure 127 shows an example:

BROWSE - PARTS LIST FOR ISPFPROJ.ABL.PLI(*)									
COMMAND ==> _					SCROLL ==> PAGE				
FROM	VIA	FROM	VIA	MEMBER	TO	VIA	TO	VIA	
***** TOP OF DATA *****									
				(MEMBERA)					
		MEMBERB	C	(MEMBERB)	MEMBERC	C	MEMBERE	I	
				(MEMBERC)	MEMBERD	C			
					MEMBERG	C*			
		MEMBERC	C	(MEMBERD)	MEMBERE	I			
MEMBERC	I	MEMBERD	I	(MEMBERE)					
				(MEMBERF)					
***** BOTTOM OF DATA *****									

Figure 127. Member Parts List Display

The figure shows that:

- Library ISPFPROJ.ABL.PLI contains members:

MEMBERA Has no calls or includes.

MEMBERB Calls MEMBERC.

MEMBERC Calls MEMBERD and MEMBERG, and includes MEMBERE. The asterisk (*) beside the C in the fourth VIA column means that MEMBERG was not found in the input library.

MEMBERD Includes MEMBERE.

MEMBERE Has no calls or includes.

MEMBERF Has no calls or includes.

Foreground - Member Parts List (Option 4.12)

- A parts list is requested for all members in the first data set.

You can scroll the output up or down using the scroll commands. All the Browse commands are available to you. When you finish browsing the listing, enter the END command and continue with step 17 on page 209.

- b. Option 2 (Write member parts data set) produces an intermediate sequential member parts list in the data set you named in step 5 on page 246. This data set can be either a sequential data set or a member of a partitioned data set.

If the data set has not been allocated, option 2 allocates it with a logical record length (LRECL) of 17, a block size (BLKSIZE) of 3009, and a record format (RECFM) of FB. The format of the records is shown in the following table:

Field Name	Format	Description
Member name	CHAR(8)	Subject member.
Called by or calls member name	CHAR(8)	Referenced member.
Call flag	BIT(1)	Found on a CALL statement.
Include flag	BIT(1)	Found by INCLUDE or COPY.
Not found flag	BIT(1)	Referenced member not found.
From flag	BIT(1)	Subject member called from referenced member.
To flag	BIT(1)	Referenced member called from subject member.
COBOL flag	BIT(1)	Member referenced outside valid COBOL division.
Reserved	BIT(2)	Field that is reserved.

Figure 128. Foreground Member Parts List Record Formats

9. You can do one of the following:
 - Enter other parameters and invoke the same processor.
 - Enter the END command to return to the Foreground Selection Panel and select another processor.
 - Enter the RETURN command to go to the ISPF/PDF Primary Option Menu.
 - Use the jump function (=) to choose any primary option.

Member Not Found

A *primary library* is one of the number of libraries specified in the GROUPS FOR PRIMARY MEMBERS field. A *primary member* is a member that starts the member parts explosion chain. An *explosion chain* is the order in which members are nested, starting with the primary member and continuing through each member that it includes, calls, or copies.

The chain is broken when a member cannot be found in the set of concatenated libraries or no more members are referenced. If a member cannot be found, the name is flagged with an asterisk (*) and processing continues. For instance, internally-called routines are not found.

When no more primary members can be found, the listing is printed, written, or browsed. Calls to internal routines or variable names result in the "member not found" flag being set.

Foreground - Member Parts List (Option 4.12)

CHAPTER 9. COMMAND (Option 6)

This chapter describes Command, option 6 on the ISPF/PDF Primary Option Menu. The Command option allows you to enter a TSO command or a CLIST specification.

Chapter 9. Command (Option 6)

When you select this option, the TSO Command Processor panel shown in Figure 152 is displayed. PDF allows you to enter TSO commands and CLISTs in the COMMAND field of any panel and in the line command field on data set list displays (option 3.4). However, the TSO Command Processor panel provides two additional capabilities. On this panel, you can enter:

- A long command that continues on the following two lines
- Session Manager mode, but only if this licensed program is installed and if your installation chooses to display the ENTER SESSION MANAGER MODE field on the TSO Command Processor panel.

```

----- TSO COMMAND PROCESSOR -----
ENTER TSO COMMAND OR CLIST BELOW:

---> _

ENTER SESSION MANAGER MODE ---> NO (YES or NO)
  
```

Figure 152. TSO Command Processor Panel

The TSO Command Processor panel contains the following fields:

ENTER TSO COMMAND OR CLIST BELOW:

Enter a command or CLIST to the right of the arrow, using standard TSO syntax. You can continue it on the next two lines, if necessary, to a maximum of 230 characters. You can also enter ISPF commands, such as RETURN or END, in this field.

See “Entering TSO Commands and CLISTs” on page 276 if you need more information.

Entering TSO Commands and CLISTs

ENTER SESSION MANAGER MODE

If the Session Manager licensed program is installed and available, use this field to tell PDF whether you want to invoke it. The valid values are:

YES	Tells PDF to invoke Session Manager.
NO	Tells PDF not to invoke Session Manager.

See "Using the Session Manager" on page 277 if you need more information.

Entering TSO Commands and CLISTs

You do not need to enter TSO before the command on this panel as you do on other panels, unless the command exists in both ISPF and TSO and you want to execute the TSO command. If you use TSO, your executed command is blanked out when the TSO Command Processor panel is displayed again.

For example, the HELP and PRINT commands are interpreted as the ISPF HELP and PRINT commands, unless you precede them with TSO. Therefore, to get TSO HELP information, enter:

===>

Rules for Entering TSO Commands

Do *not* enter the following commands under PDF:

- LOGON and LOGOFF
- ISPF, PDF, or ISPSTART
- TEST
- Commands that are restricted by TSO or PDF
- Commands started through the Authorized Program Facility (APF), such as Resource Access Control Facility (RACF) commands, unless the TSO/E Release 3 licensed program is installed
- Commands that invoke an APF-authorized program
- ISPEXEC service calls.

Rules for Entering CLISTs

You can enter a CLIST name on this panel, but the following restrictions apply:

- The CLIST cannot invoke the restricted commands shown in the preceding list. However, this does not apply to ISPEXEC, which can be invoked in a CLIST.
- CLIST attention exits may produce unwanted results when you press PA1.
- CLIST error exits are not entered for ABENDs.
- TERMIN command procedure statements may cause unwanted results.

Note: A command issued via its synonym may cause unwanted results.

Using the Session Manager

If the Session Manager licensed program is installed and available, the TSO Command Processor panel includes this field:

ENTER SESSION MANAGER MODE ===> (YES or NO)

If you specify YES in this field, any display output is displayed in the Session Manager TSOOUT stream.

Note: If graphics interface mode is active, Session Manager does not get control of the screen. Graphics interface mode is activated when a GRINIT service has been issued, but a GRTERM service has not been issued. See Dialog Management Services and Examples for more information about these two services.

The PF key definitions are not transferred to the Session Manager from PDF. When the command ends, the Session Manager prompts you to enter a null line to return to PDF control and displays the TSO Command Processor panel again when you do.

If you specify NO, terminal I/O occurs as though the Session Manager were not installed. The terminal operates in normal TSO fashion. Any communication with the command is in line-I/O mode. When the command ends, three asterisks (***) are displayed. Press the ENTER key to display the TSO Command Processor panel again in full screen mode.

To interrupt a TSO command or CLIST, press the PA1 key. The TSO command ends and the TSO Command Processor panel is displayed again. If terminal input is inhibited, press the RESET key before pressing the PA1 key. If you are in Session Manager mode, enter a null line to return to ISPF full screen mode.

When the TSO Command Processor panel is displayed again, the command that was just executed is displayed to the right of the arrow. Enter another

Using the Session Manager

command, or the END command to return to the ISPF/PDF Primary Option Menu.

For terminals with primary and alternate screen sizes, ISPF does not check to make sure the same screen settings are in effect when a command or CLIST ends. If you invoke a CLIST or command that changes the screen settings, you are responsible for saving and restoring them before control is returned to ISPF.

CHAPTER 10. DIALOG TEST (Option 7)

This chapter describes Dialog Test, option 7 on the ISPF/PDF Primary Option Menu. The Dialog Test facility is a powerful tool for dialog developers. It allows you to test entire ISPF applications, as well as incomplete ISPF dialogs that you are in the process of developing.

The chapter includes information about the Dialog Test environment, variable usage, and severe error handling. Also, the chapter describes the set of Dialog Test primary commands and line commands, which are valid only when you are using the Dialog Test option.

The chapter concludes with a complete discussion of the 10 options you can select from the Dialog Test Primary Option Menu.

Chapter 10. Dialog Test (Option 7)

The Dialog Test option (7) provides you with facilities for testing both complete ISPF applications and ISPF dialog parts, including functions, panels, variables, messages, tables, and skeletons. The Dialog Test option allows you to:

- Invoke selection panels, command procedures, and programs
- Display panels
- Add new variables and change variable values
- Display a table's structure and status
- Display, add, modify, and delete table rows
- Browse the ISPF log
- Execute dialog services
- Add, modify, and delete function and variable trace definitions
- Add, modify, and delete breakpoint definitions.

You can use TSO TEST to complement this option if you want to examine and manipulate non-ISPF storage areas.

When testing a dialog, you usually do so in one of two ways:

- Test individual dialog parts, including panels, skeletons, and messages, without invoking a function or a selection panel. Eventually, you end your test session by entering the END command on the Dialog Test Primary Option Menu.
- Test dialog functions, including programs, commands, and selection menus, using the Functions option (7.1). You can define traces and breakpoints before invoking the function.

Any requested traces for variable usage and dialog service calls are written to the ISPF log. You can browse the log using the Log option (7.5).

If you define a breakpoint and the function gets to it, dialog execution is suspended and Dialog Test displays the Breakpoint Primary Option Menu (Figure 173 on page 323). At this point, you can access and manipulate dialog parts, such as variables, tables, and so forth. Then, if you select the Go option from the Breakpoint Primary Option Menu, the dialog resumes execution.

At completion, you are returned to the Functions option (7.1). If you select the Cancel option from the Breakpoint Primary Option Menu, the dialog is cancelled and the first primary option panel that you were shown during your terminal session is displayed again. For example, if the first screen displayed when you began your session was a master application panel that is different from the ISPF/PDF Primary Option Panel, that master application panel is displayed again.

Environment

The Dialog Test Primary Option Menu, Figure 153, follows the conventions for a primary option panel. If you use the RETURN command from one of the selected Dialog Test options, the Dialog Test Primary Option Menu is displayed again. If you use the END command from this panel, you return to the ISPF/PDF Primary Option Menu.

When you enter Dialog Test, you enter a new user application with an application ID of ISR. All options listed on the Dialog Test Primary Option Menu operate in this context. If you invoke a new function using the Functions option (7.1), a SELECT service call is performed, and the rules for the SELECT service are followed.

```
----- DIALOG TEST PRIMARY OPTION MENU -----
OPTION ===> _

 1 FUNCTIONS      - Invoke dialog functions/selection menus
 2 PANELS         - Display panels
 3 VARIABLES      - Display/set variable information
 4 TABLES        - Display/modify table information
 5 LOG            - Browse ISPF log
 6 DIALOG SERVICES - Invoke dialog services
 7 TRACES         - Specify trace definitions
 8 BREAKPOINTS    - Specify breakpoint definitions
 T TUTORIAL       - Display information about Dialog Test
 X EXIT          - Terminate dialog testing

Enter END command to terminate dialog testing.
```

Figure 153. Dialog Test Primary Option Menu

Dialog Test is itself a dialog and, therefore, uses dialog variables. Since it is important to allow your dialog to operate without interference, as though in a production environment, Dialog Test accesses and updates variables independently of your dialog variables.

All breakpoints and traces that you set in Dialog Test exist only while you remain within the Dialog Test option.

You should *always* allocate the ISPF log when using Dialog Test. Do not suppress its generation by entering 0 in the Primary pages field used with the Log/List option (0.2). Dialog Test writes trace data to the log when you request it. Also, if Dialog Test finds an unexpected condition, it writes problem data to the log.

When you enter Dialog Test, you are given the following ISPF facilities:

- All functions you normally get by specifying the TEST parameter on the PDF command

- The logging of all severe errors, both from user dialogs and Dialog Test. This is normally done when you specify TRACE or TRACEX on the PDF command.
- The suspension of the logging of all ISPEXEC dialog service requests. Such logging normally occurs when you specify TRACE or TRACEX on the PDF command. You should use the Traces option (7.7).

These facilities become active for all logical screens when you are using split-screen mode and remain active while you remain in PDF, even after you leave Dialog Test. Logging of all ISPEXEC service requests is resumed when you leave Dialog Test if TRACE or TRACEX is specified on the PDF command. The TEST, TESTX, TRACE, and TRACEX options are described in *Dialog Management Guide*.

Variable Usage

When you select the Dialog Test option, you are given a new function variable pool, a new shared variable pool, and the ISRPROF profile pool. These pools are used if you set a variable, display a panel, call an ISPF service, and so forth. When you invoke a new dialog, Dialog Test uses the SELECT service, and follows the rules for the creation of new variable pools. For example, if you invoke a new dialog using the NEWPOOL option, Dialog Test creates new shared and function variable pools for you. The profile variable pool, ISRPROF, remains as it was.

If you set a dialog variable in the shared pool from a dialog running under Dialog Test and then invoke the dialog again from the COMMAND line, you cannot retrieve the value of that variable.

Dialog variables should be initialized and set in the context of the dialog's processing. A dialog's function variable pools are created when it is invoked; that is, when the SELECT is done. Therefore, to set function variables in newly created pools, you must define a breakpoint early in your dialog's processing, at a point *before* the function is invoked.

For example, if you invoke a dialog with the NEWPOOL parameter, you must define a breakpoint in the dialog before the first function is invoked to access that dialog's function and shared variable pools. You can change the dialog's profile variable pool before invoking the dialog, since a new profile variable pool is not created.

When your dialog terminates, all variable pools created at dialog invocation are deleted.

Note: ISPF does not support TSO global variables. You may find a severe dialog test error when testing a dialog that references a global variable.

Severe Error Handling

If your dialog finds a severe error when it invokes a dialog service, the error is handled as requested by the dialog. The current CONTROL service ERRORS setting, CANCEL or RETURN, determines what is done. If CANCEL is in effect, you can choose whether to continue dialog testing when the Error Message panel is displayed.

Note: If you choose not to continue dialog testing, you return to the ISPF/PDF Primary Option Menu, but test mode is still active.

If you find a severe error when manipulating your dialog at a breakpoint, Dialog Test assumes that the CONTROL service ERRORS setting is CANCEL. For example, if you display a panel at a breakpoint and that panel is not found, the Error Message panel is displayed. This occurs even if your current dialog has an ERRORS setting of RETURN.

Regardless of the ERRORS setting, all your severe errors are logged.

If Dialog Test finds a severe error during its processing, the details are logged and the following message is shown to you on an error message display:

***** ERROR IN THE ISPF LOG *****

Dialog Test errors can occur because:

- Proper ISPF libraries are not being used.
- A programming problem has been encountered.
- You have attempted to execute Dialog Test recursively.
- You have invoked a Dialog Test option without being in test mode, or without invoking Dialog Test first.

You should browse the ISPF log to find the problem.

Commands

You can enter ISPF and PDF primary commands on Dialog Test panels. Five commands have special meaning during Dialog Test operations. You enter them in the COMMAND line of the applicable Dialog Test option

panel. These commands, and the Dialog Test options with which they function, are:

Primary Command	Valid Option(s)
CANCEL	Variables (Option 7.3) Tables (Option 7.4) ¹ Traces (Option 7.7) ² Breakpoints (Option 7.8)
END	Variables (Option 7.3) Tables (Option 7.4) ³ Traces (Option 7.7) ² Breakpoints (Option 7.8)
LOCATE	Variables (Option 7.3) Tables (Option 7.4) ⁴ Traces (Option 7.7) ² Breakpoints (Option 7.8)
QUAL	Breakpoints (Option 7.8)
RESUME	Breakpoints (Option 7.8)

Dialog Test has three line commands that have special meaning during testing operations. These commands, and the options with which they function, are:

Line Command	Valid Options
D (delete)	Variables (Option 7.3) Tables (Option 7.4) ¹ Traces (Option 7.7) ² Breakpoints (Option 7.8)
I (insert)	Variables (Option 7.3) Tables (Option 7.4) ¹ Traces (Option 7.7) ² Breakpoints (Option 7.8)
R (repeat)	Tables (Option 7.4) ¹ Traces (Option 7.7) ² Breakpoints (Option 7.8)

¹ Valid only with Tables options 3 and 4.

² Valid only with Traces options 1 and 2.

³ Valid only with Tables options 1, 3, and 4.

⁴ Valid only with Tables options 1, 3, 4, and 5.

Dialog Test - Commands

When using the Dialog Test primary and line commands, you should be aware of the following:

- You can specify both a primary command and line commands before you press the ENTER key.
- You can enter multiple line commands on the display.
- You cannot carry out a deletion if one of the included lines contains another line command.
- You can delete lines that contain an input error.
- The line commands are processed in row order when you press the ENTER key. Any fields changed in the row are handled *before* a line command is processed.
- A primary command is handled *after* processing for all line commands is complete.
- As in the editor, you can specify a number with each line command to denote repetitive operation, unless you are using the Variables option (7.3). To avoid conflict with the I (insert) line command, the Variables option does *not* let you enter the D*n* form of the D (delete) line command, where *n* is the number of lines to be deleted. Therefore, enter a single D line command on each line you want to delete. You can enter this command on more than one line before pressing the ENTER key.

Ending the Current Option Without Saving Changes

The CANCEL command ends the current option. Any changes made to the data are ignored.

Saving Changes

The END command ends the current option. Any changes made to the data now take effect.

Finding a Character String

The LOCATE command searches for a character string and positions a scrollable display to the next row that contains the string. The scan starts at the end of the first row currently being displayed. A message is displayed indicating the result of the scan.

The format of the LOCATE command is:

LOCATE string

where:

string The character string you are trying to find. If the string ends in an asterisk (*), a scan for the characters preceding the asterisk is done.

Displaying Breakpoint Qualification Data

The QUAL command can only be entered from the Breakpoints panel. It displays the breakpoint qualification data.

Restoring the Format of the Breakpoints Panel

The RESUME command is entered on the Breakpoints panel when qualification parameter values are shown. It restores the format of the Breakpoints panel. Each breakpoint that has qualification is flagged by the characters ***QUAL*** in columns 75 to 80 on that line of the Breakpoints panel.

Dialog Test Line Commands

The following line commands have special meaning during testing operations:

D - Deleting Lines

The D command deletes one line or n lines starting with this line. The format is:

D

D n

If you are using the Variables option (7.3), the second form of the D line command does not apply. To avoid conflict with the I (insert) line command, the Variables option does *not* let you enter D n , where n is the number of lines to be deleted. Therefore, enter a single D line command on each line you want to delete. You can enter this command on more than one line before pressing the ENTER key.

I - Inserting Lines

The I command inserts one line or n lines directly after this line, with underscores and quotation marks in the appropriate fields. The format is:

I

I n

Dialog Test - Commands

R - Repeating Lines

The R command repeats this line once or n times. The format is:

R

Rn

Functions (Option 7.1)

The Functions option (7.1) allows you to test a dialog function without having to build supporting code or panels. Dialog functions include panels, command procedures, and programs. The name of the dialog function and the parameters you can pass are the same as those that you can specify from a dialog function when you invoke the SELECT service. When you press the ENTER key, a SELECT is performed.

If you invoke a new function or selection panel at a breakpoint, the previous function or selection panel is suspended and the new one executed. When the new activity finishes, the Invoke Dialog Function/Selection Menu is displayed. The old activity resumes when you enter the END command. When the originally-invoked function completes execution, the Invoke Dialog Function/Selection Menu is displayed again.

System programmers can use the Functions option to run the LMF migration utility. This utility converts Version 2 Release 2 LMF control files to the Version 2 Release 3 format. See *Installation and Customization* for more information.

When you select the Functions option, the panel in Figure 154 is displayed to allow you to specify the dialog function that you want to test.

```

----- INVOKEDIALOGFUNCTION/SELECTIONMENU -----
COMMAND ---->

INVOKESLECTIONMENU:
  PANEL ----> _ OPT ---->

INVOKECOMMAND:
  CMD ---->
  LANG ----> (APLORBLANK)
  MODE ----> (LINE, FSCR, ORBLANK)

INVOKEPROGRAM:
  PGM ----> PARM ---->
  MODE ----> (LINE, FSCR, ORBLANK)

NEWAPPL ----> NO ID ---->
NEWPOOL ----> NO PASSLIB ----> NO
    
```

Figure 154. Invoke Dialog Function/Selection Menu

To invoke a function, you must specify a value for either the PANEL, CMD, or PGM field.

Note: You cannot specify more than one of these fields.

- To invoke a selection panel, use the following fields under the heading INVOKESLECTIONMENU:

PANEL The name of the selection panel to be displayed.

Dialog Test - Functions (Option 7.1)

OPT An optional parameter indicating the first selection option that must be valid from the specified selection panel. This input field continues onto the next line on the panel.

- To invoke a command, use the following fields under the heading INVOKE COMMAND:

CMD The name of a command procedure written in CLIST, or any TSO command, to be invoked as a dialog function. You can include command parameters.

Use the “%” prefix symbol to tell PDF to remove the Invoke Dialog Function/Selection Menu and use the full screen to display the results of a CLIST invocation. Three asterisks show CLIST completion. Press the ENTER key to return to the Invoke Dialog Function/Selection Menu.

If you omit the “%” prefix, PDF interprets the command as a TSO command, using line mode to display the command results at the bottom of the Invoke Dialog Function/Selection Menu.

LANG An optional parameter. Enter APL to specify the use of the APL language. If this is your first APL request during the session, the command specified in the CMD keyword is invoked and an APL2 environment is established. If this is not your first APL request during this session, the string specified after the CMD keyword is passed to the APL2 workspace and executed.

To specify any language other than APL, leave this field blank.

MODE An optional parameter that overrides:

- Automatic line mode entry, caused when a TSO command is entered.
- Automatic full-screen display caused by the “%” CLIST prefix. However, it does not prevent PDF from invoking the command as a CLIST.

If you leave this field blank, the “%” prefix has its normal effect. The valid values for this field are:

LINE Used to enter line mode when invoking a CLIST.

FSCR Used to enter full-screen mode when invoking a TSO command.

- To invoke a program, use the following fields under the heading **INVOKE PROGRAM**:

PGM	The name of a program to be invoked as a dialog function.				
PARM	Optional parameters to be passed to the program. This input field continues onto the next line on the panel.				
MODE	An optional parameter used to tell PDF whether to display the program results in line mode or full-screen mode. If you leave this field blank, PDF uses line mode as the default. The valid values for this field are: <table><tr><td>LINE</td><td>Used to enter line mode when invoking a program. Results of the program are displayed at the bottom of the Invoke Dialog Function/Selection Menu.</td></tr><tr><td>FSCR</td><td>Used to enter full-screen mode when invoking a program. PDF removes the Invoke Dialog Function/Selection Menu and uses the full screen to display the program results. Three asterisks show program completion. Press the ENTER key to return to the Invoke Dialog Function/Selection Menu.</td></tr></table>	LINE	Used to enter line mode when invoking a program. Results of the program are displayed at the bottom of the Invoke Dialog Function/Selection Menu.	FSCR	Used to enter full-screen mode when invoking a program. PDF removes the Invoke Dialog Function/Selection Menu and uses the full screen to display the program results. Three asterisks show program completion. Press the ENTER key to return to the Invoke Dialog Function/Selection Menu.
LINE	Used to enter line mode when invoking a program. Results of the program are displayed at the bottom of the Invoke Dialog Function/Selection Menu.				
FSCR	Used to enter full-screen mode when invoking a program. PDF removes the Invoke Dialog Function/Selection Menu and uses the full screen to display the program results. Three asterisks show program completion. Press the ENTER key to return to the Invoke Dialog Function/Selection Menu.				

You can also specify:

NEWAPPL	Indication of whether a new application is being invoked. <table><tr><td>YES</td><td>This function is a new application.</td></tr><tr><td>NO</td><td>This function is not a new application.</td></tr></table>	YES	This function is a new application.	NO	This function is not a new application.
YES	This function is a new application.				
NO	This function is not a new application.				
ID	A one- to four-character ID for a new application. If you invoke a new application and leave the ID field blank, the default ID of ISR is used. Note that the ID determines the names of the profile and the command table to be used for the application.				
NEWPOOL	Indication of whether a new shared variable pool is to be created. This value is ignored if the value for NEWAPPL is YES. <table><tr><td>YES</td><td>Create a new shared variable pool.</td></tr><tr><td>NO</td><td>Do not create a new shared variable pool.</td></tr></table>	YES	Create a new shared variable pool.	NO	Do not create a new shared variable pool.
YES	Create a new shared variable pool.				
NO	Do not create a new shared variable pool.				

PASSLIB Shows that the current set of application-level ISPF libraries, if any sets exist, is to be used by the application being selected. You can specify PASSLIB only if you also specify YES in the NEWAPPL field.

YES Pass the current set of application library definitions to the new application.

NO Do not pass the current set of application library definitions to the new application.

Note: For more information about the PASSLIB field, see the description of the SELECT service in Dialog Management Services and Examples.

Panels (Option 7.2)

When you are developing panels, you can use the Panels option (7.2) to test newly created or changed panels and messages without having to build supporting code to display them. Any variables referenced and set during panel processing are written to the current function variable pool. When you select the Panels option (7.2), the panel in Figure 155 is displayed.

----- DISPLAY PANEL -----

COMMAND ===>

PANEL NAME ===> _

MESSAGE ID ===> (Optional)

CURSOR FIELD ===> (Optional)

CURSOR POSITION ===> (Optional)

Figure 155. Display Panel

On the panel are the following fields:

PANEL NAME

The name of the panel to be displayed.

MESSAGE ID

The identifier of a message to be displayed on the panel.

CURSOR FIELD

The name of the field on the panel where the cursor is to be positioned.

CURSOR POSITION

An integer specifying the position in the field where the cursor is to be placed.

If you specify a panel name, the MESSAGE ID, CURSOR FIELD, and CURSOR POSITION fields are optional.

These are the same parameters that a dialog function can specify when invoking the DISPLAY service.

When the panel is displayed, the)INIT and)PROC sections of the panel are processed in the same way the DISPLAY service would process them.

If you want to set variables before you display the panel, you can use the Variables option (7.3) to do so. When you display the panel, you can enter

Dialog Test - Panels (Option 7.2)

new or changed data and then verify the variables by using the Variables option (7.3) again. Data that you enter on the panel you display is retained until you change it, leave Dialog Test, or reset the function pool.

Figure 156 shows the message that is displayed if you specify a message ID and, optionally, a cursor position without identifying a panel name. The long message portion of the identified message is displayed when you enter the HELP command on that panel.

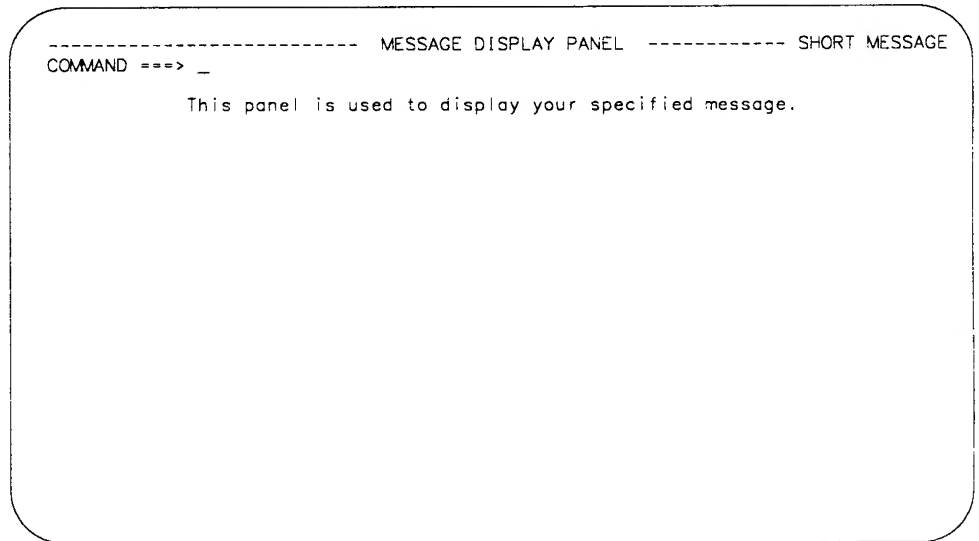


Figure 156. Message Display Panel

When you enter the END command from the panel being tested, the Display Panel reappears on the screen.

P	Pool that contains the variable; a required one-character field, where:
	V Function pool; the variable was defined with the VDEFINE service.
	I Function pool; a variable that was created by a CLIST or by using the Variables option. This is an “implicit” variable, which was not explicitly defined by using the VDEFINE service.
	S Shared variable pool.
	P Profile variable pool.
A	Attributes of the variable, a non-modifiable one-character field, where:
	N Non-modifiable variable. Some system-reserved variables are not modifiable.
	T Truncated variable value. The actual value is longer than the 58 characters that you can display on the panel. If you change a line showing a truncated value, only the data shown on the display is stored as the new value.
VALUE	Value of the variable. The value can be up to 58 characters in length.

Variables Commands

The Variables option (7.3) uses the CANCEL, END, and LOCATE commands, and the I (insert) and D (delete) Dialog Test line commands. “Commands” on page 282 describes these commands.

Normally, the variable pools are updated with the data from the display when you use the END command to leave the option.

Manipulating Variables

The rows of the display are sorted in the following order:

1. By pool (function, then shared, then profile)
2. By function pool type (V then I)
3. Alphabetically by variable name within each pool.

Insertions are left where they are entered on the display. Changes to the display are processed when you press the ENTER key. Updating of the variable pools occurs when you enter the END command.

Creating New Variables

You can create new dialog variables, but you cannot create two variables with the same name in the same variable pool.

To create a new variable, you can do one of two things:

- Use the I line command to insert a new row, and then enter the variable name, pool, and value on the new line. For each field, move the cursor to the start of the field and enter new information. The underscores are pad characters; you do not need to blank them out.
- Type over the name of an existing variable, its pool indicator, or both. This creates a new variable and resets the old variable's value to nulls.

If you change a truncated value, the undisplayed portion is lost. The new variable value consists only of the visible portion on the screen.

New function pool variables are given an I (implicit) attribute and a CHAR format. If you enter "F" in the attribute field, PDF changes it to "I."

By using the second method, you can interchange the values of two or more variables by simply changing their names. For example, you can interchange the values for variables A and B by changing the variable name A to B and name B to A, and then pressing the ENTER key.

Deleting Variables

Any dialog variable in the shared and profile pools can be deleted, unless it has an N attribute. Though you cannot delete a variable from the function pool, you can set its value to blanks.

To delete a variable, use the D line command. However, to avoid conflict with the I (insert) line command, the Variables option does *not* let you enter Dn , where n is the number of lines to be deleted. Therefore, enter a single D line command on each line you want to delete. You can enter this command on more than one line before pressing the ENTER key.

Variables Usage Notes

When using the Variables option (7.3), you should be aware of the following:

Input errors

Correct any errors before leaving a display. If you cannot correct the errors, use the CANCEL command.

Length and format errors in variables defined with the VDEFINE service are detected when you enter the END command. If ISPF finds such an error, it prompts you to fix the variable value.

Dialog Test - Variables (Option 7.3)

Test mode

Variable manipulations carried out under Dialog Test at a breakpoint are considered an extension of your dialog and, as such, are handled in user mode. Dialog variables, table data, and service return codes that you introduce, delete, or change are treated as though your dialog had made those changes.

Variable life

Profile variables that you create remain in your profile pool from one Dialog Test session to another. Shared and function variables exist only for the duration of Dialog Test.

Split-screen mode

In split-screen mode, two logical screens can share a profile variable pool. Since the Variables option (7.3) takes a snapshot of the variables, any change to a profile variable on one screen is not immediately reflected on the other screen. To get the latest changes, select the Variables option (7.3) again. Also, when one profile variable is changed on two logical screens using split-screen mode, the changed profile variable on the screen where the last END command was entered takes precedence.

Variable value

Variables defined with the VDEFINE service as non-character are displayed in translated form. Any changes made to the variable's value should conform to the defined format.

Do not change them using the hexadecimal representation. A format or length error causes a message to be displayed when you use the END command. When a VDEFINE error occurs, a panel identifies the data and its value and describes the error. You must then correct the error and press the ENTER key. If you create a new variable by changing the pool indicator of an existing variable defined as non-character, the new variable has character (CHAR) format.

Hexadecimal data

Hexadecimal data that cannot be displayed is translated to displayable characters or entered using the form:

X'nnnnnnnn'

where:

n An integer 0 through 9 or an alphabetic character A through F. There must be an even number of characters within the quotation marks.

DBCS data

A variable defined as DBCS by the VDEFINE service or displayed through the field with FORMAT(DBCS) specified in the test environment is displayed using the form:

'¬[DBDBDB]'

where:

[and] Represent the SO (shift-out) and SI (shift-in) characters, respectively.

If you enter a DBCS value in this format on the Variable Display and Set panel, only the DBCS characters are stored.

Tables (Option 7.4)

The Tables option (7.4) allows you to examine and manipulate the rows of a table, and to display table structure and status. When you select this option, the panel in Figure 158 is displayed, on which you show the table function you want and the parameters needed to identify the table.

```
----- TABLES -----
OPTION ----> _

  1 Display row          4 Add row
  2 Delete row          5 Display structure
  3 Modify row          6 Display status

TABLE NAME      ---->

ROW IDENTIFICATION:
  BY ROW NUMBER ----> *          (* = current row)
  BY VARIABLE    VALUE          (Search for row if row number blank)
  _____
  _____
  _____
  _____
  _____

DBCS COLUMN SPECIFICATION:
  _____  _____  _____  _____  _____  _____
```

Figure 158. Tables Panel

Use the fields on the Tables panel as follows:

OPTION Enter the number of one of the functions displayed on the Tables panel.

TABLE NAME
Enter the name of the table you are interested in. The table must be open for all but the Display Status option (6) of the Tables option (7.4). If you need to open the table, use the Dialog Services option (7.6).

CURRENT ROW
Displays the position number of the current row after you have identified a table. This field is non-modifiable.

ROW IDENTIFICATION
Identify a row, either directly by row number or indirectly by specifying a search argument of table variable names and their values.

BY ROW NUMBER
Enter the position number of the table row that you want, or enter * for the current row. If you are adding a row, you can use:

TOP Make the new row first in the table.
BOTTOM Make the new row last in the table.

BY VARIABLE

Type over the underscores beneath this heading with the names of variables whose values are to be used to search the table for a row with matching contents.

VALUE

Enter the value to be used in the search, up to 58 characters. For an abbreviated search, enter the beginning characters followed by an asterisk.

You can specify a DBCS value in the form:

¬ [DBDBDB]

where:

[and] Represent the SO (shift-out) and SI (shift-in) characters, respectively. For an abbreviated search, enter a two-byte asterisk (*) at the end of the DBCS value. For example:

¬ [DBDB**]

where:

** Represents the two-byte asterisk character.

DBCS COLUMN SPECIFICATION

Enter the variable names of the values that are DBCS data. The value of the variable is displayed using the form:

¬ [DBDBDB]

If you enter a DBCS value in this format on the Modify Row panel or the Add Row panel, only the DBCS characters are stored, regardless of the DBCS column specification.

Once you specify a table name, it is retained until you change it or until you leave Dialog Test.

For the Display Row (1), Delete Row (2), Modify Row (3), and Add Row (4) options of the Tables option (7.4), you must identify the row you want to display, delete, modify, or add. To do this, you can specify a row number in the BY ROW NUMBER field, or you can use the BY VARIABLE and VALUE fields to specify a search argument list. To show the current row, leave the asterisk in the ROW NUMBER field. If you specify both a row number and a search argument, the row number is used and the search argument is ignored.

The current row pointer in the table can be changed only at your request or by your dialog.

The search argument list consists of variable names and values that let you specify the values that specific variables have in a row. You can specify the complete value, abbreviate the value with an asterisk to find a row containing a variable beginning with specified characters, or leave the row

Dialog Test Tables (Option 7.4)

blank. The search begins with the row following the current row. If a row matching the search argument is not found, the current row pointer is set to the top. You can, at your option, repeat the search.

The following sections describe the options shown at the top of the Tables panel.

1 - Display Row

You can use the Display Row option to display the contents of an existing row in an open table. When you select the Display Row option, do the following on the Tables panel:

- Specify the name of an open table in the TABLE NAME field.
- Specify a row number or a search argument list to identify a row.

When you press the ENTER key, you are shown the table row data on a display that you can scroll (Figure 159). In the figure, the variables constitute one table row.

```
DISPLAY ROW  TABLE SAMTB1  ROW 3  ----- ROW 3 OF 21
COMMAND ----> _                               SCROLL ----> PAGE

VARIABLE  T  A  VALUE
K1_____ K   This is the value for key K1
K2_____ K   My value for K2
K3_____ N   This is the value to be used
N1_____ N   The first name value is this field
N2_____ N   Name variable N2 is also a save variable
SAVE1____ S   Save variable 1 has this value
SAVE2____ S   Show save variable 2 this way
N2_____ S   Save variable N2 is also a name variable
***** BOTTOM OF DATA *****
```

Figure 159. Display Row Panel

Each line on the display indicates:

VARIABLE	Variable name.
T	Type of variable:
	K Key variable.
	N Name variable; non-key.
	S Save (extension) variable.
A	Attribute of each variable:
	T Truncated to 58 characters for display.
VALUE	The first 58 characters of the variable value.

Display Row Commands

The Display Row option uses the END and LOCATE commands. "Commands" on page 282 describes these commands.

2 - Delete Row

You can use the Delete Row option to remove an existing row from an open table. When you select the Delete Row option, do the following on the Tables panel:

- Specify the name of an open table in the TABLE NAME field.
- Specify a row number or a search argument list to identify a row.

When you press the ENTER key, a panel is displayed (Figure 160) to allow you to confirm the delete request.

```
----- CONFIRM TABLE ROW DELETE -----  
COMMAND ==> _  
TABLE NAME : SAMTB1  
ROW NUMBER : 3  
  
Press ENTER key to confirm delete  
Enter END command to cancel delete
```

Figure 160. Confirm Table Row Delete Panel

The fields on the panel are:

TABLE NAME Name of an open table.

ROW NUMBER Number of the row to be deleted.

Press the ENTER key to delete the row, or enter the END or CANCEL command to cancel the deletion.

3 - Modify Row

You can use the Modify Row option to change the contents of an existing row of an open table. When you select the Modify Row option, do the following on the Tables panel:

- Specify the name of an open table in the TABLE NAME field.
- Specify a row number or a search argument list to identify a row.

Dialog Test Tables (Option 7.4)

When you press the ENTER key, a display that you can scroll (Figure 161 on page 302) is shown. In this example, row 6 of table SAMTB1 is displayed for modification. Row 6 contains eight variables.

```
MODIFY ROW TABLE SAMTB1 ROW 6 ----- ROW 6 OF 21
COMMAND ----> _ SCROLL ----> PAGE

MODIFY VARIABLE VALUES AND SAVENAMES. UNDERSCORES NEED NOT BE BLANKED.
ENTER END COMMAND TO FINALIZE CHANGES.

VARIABLE T A VALUE

**** K1_____ K Value for key K1
**** K2_____ K Value for key K2
**** K3_____ K
**** N1_____ N The first name value is this field
**** N2_____ N Name variable N2 is also a save variable
**** SAVE1____ S Save variable 1 has this value
**** SAVE2____ S Show save variable 2 this way
**** N2_____ S Save variable N2 is also a name variable
**** _____ S
**** _____ S
**** _____ S
**** _____ S
**** _____ S
**** _____ S
***** BOTTOM OF DATA *****
```

Figure 161. Modify Row Panel

Each line on the panel represents a variable in row 6 of the table, and contains:

**** Line command area.

VARIABLE Variable name, modifiable only for save variables.

T Type of variable, non-modifiable:

K Key variable.

N Name variable; non-key.

S Save (extension) variable.

A Attribute of each variable, non-modifiable:

T Truncated to 58 characters for display.

VALUE Value of the variable up to 58 characters.

Enter or change the values for the key, name, and save variables in the VALUE column. Enter new save variables by typing over the underscores in the NAME column with the variable names and specifying the desired values. The underscores are pad characters; you do not need to blank them out.

When using the Modify Row option, be aware of the following:

- If the table has keys, the values for the keys in the added row must be different from those in the existing rows when you leave the Modify Row option. Otherwise, a message is displayed and the row is displayed again so you can change the keys.

- If the table was sorted using the TBSORT dialog service and a sort field is modified, the row's position in the table may change to preserve the search order.
- You cannot change the variable name for a key variable or name variable; if you do, an error message is displayed and the original name is restored.
- You cannot delete a key or name variable and its value from the display or table row.
- If you delete a save variable, assume that the variable no longer exists in this row.
- If more than one variable entry has the same name, all instances of that variable are assigned the value of the last occurrence of the variable; that is, the occurrence closest to the bottom of a display that you can scroll.
- Blank save names are ignored and do not need to be deleted, even if data is left in the value.
- Hexadecimal data that usually cannot be displayed is translated to displayable characters or entered using the form:

X'nnnnnnnn'

where:

n An integer 0 through 9 or an alphabetic character A through F. There must be an even number of characters within the quotation marks.

- Variables defined with the VDEFINE service as non-character are shown in translated form; do not change them by using the hexadecimal representation. A format or length error causes a message to be displayed when you use the END command.
- When you leave the Modify Row option by using the END command, the row is replaced and the message **ROW MODIFIED** is issued.

Modify Row Commands

The Modify Row option uses the CANCEL, END, and LOCATE commands, and the D (delete), I (insert), and R (repeat) Dialog Test line commands. "Commands" on page 282 describes these commands. Inserted and repeated lines always have a type of S, because you can add only save variables to a row of an existing table. Also, you can delete only save variables.

4 - Add Row

- Specify the name of an open table in the TABLE NAME field.
- Specify a row number or a search argument list to identify a row.

Figure 162. Add Row Panel

Each row of the display contains:

Enter the values for the key and name variables in the **VALUE** column, which is originally initialized to all nulls. You cannot change the names of the key and name variables because they were established when the table was created.

NW: 15457 DocId: 70002077 Page 670

You can add a row with no values to the table, but you are asked to confirm such an action to guard against inadvertent use of the END command.

When using the Add Row option, be aware of the following points:

- The position of the new row in the table depends on whether the table was previously sorted using the TBSORT dialog service. If the table was sorted, the new row is placed in sort order; if it has not been sorted, the new row is placed after the row you specified.
- You cannot delete a key or name variable and its value from the display or table row.
- You cannot change the variable name for a key or name variable; if you do, an error message is displayed and the original name is restored.
- If more than one variable entry has the same name, all instances of that variable are assigned the value of the last occurrence of the variable; that is, the occurrence closest to the bottom of the display that you can scroll.
- If the table has keys, the values for the keys in the added row must be different from those in all the existing rows when you leave the Add Row option. Otherwise, a message is displayed and the row is displayed again so you can change the keys.
- Blank save names are ignored and do not need to be deleted, even if data is left in the value.
- Hexadecimal data that usually cannot be displayed is translated to displayable characters or entered using the form:

X'nnnnnnnn'

where:

- n** An integer 0 through 9 or an alphabetic character A through F. There must be an even number of characters within the quotation marks.
- Variables defined with the VDEFINE service as non-character are shown in translated form; do not change them by using the hexadecimal representation. A format or length error causes an error message to be displayed when you use the END command.

Add Row Commands

The Add Row option uses the CANCEL, END, and LOCATE commands, and the D (delete), I (insert), and R (repeat) Dialog Test line commands. "Commands" on page 282 describes these commands. Inserted and repeated lines always have a type of S, because you can add only save variables to a row of an existing table. Also, you can delete only save variables.

5 - Display Structure

When you select the Display Structure option on the Tables panel, you are shown a display of the table structure for the table specified in the TABLE NAME field. You can scroll this display (Figure 163) using the scroll commands. The table name appears in the panel header.

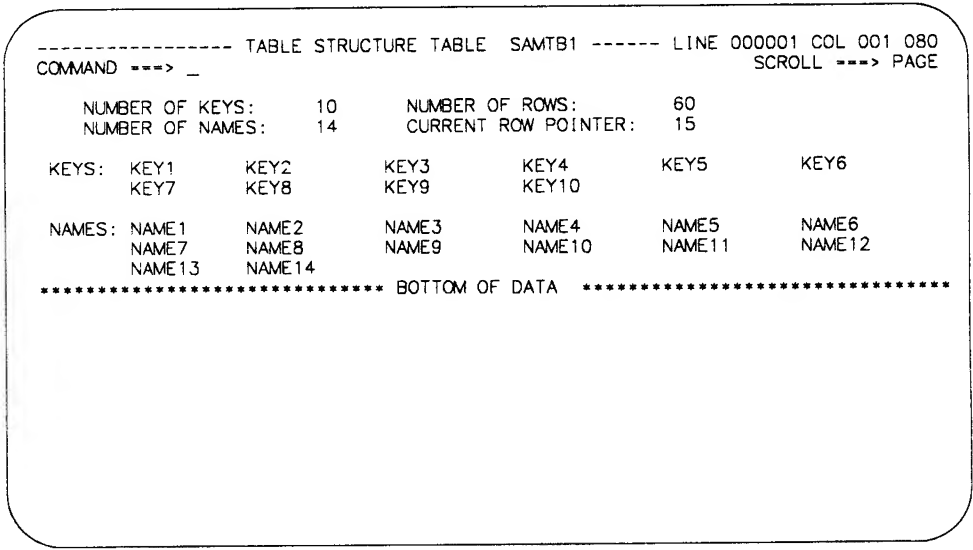


Figure 163. Table Structure Panel

The display shows the following fields:

- NUMBER OF KEYS**
Number of key variables in a row.
- NUMBER OF NAMES**
Number of name variables in a row.
- NUMBER OF ROWS**
Number of rows currently in the table.
- CURRENT ROW POINTER**
Current row pointer value.
- KEYS**
A list of the names of all the key variables.
- NAMES**
A list of the names of all the name variables.

Display Structure Command

The KEYS and NAMES lists can be scrolled, and you can use the LOCATE command to find a specific variable name. See “Finding a Character String” on page 284 for information on its use.

6 - Display Status

If you select the Display Status option from the Tables panel, one of two data information panels is displayed for the table specified in the TABLE NAME field. The information reflects all operations using the specified table, including those done at your request by the Tables options under Dialog Test.

Table Not Open

If the table is not open for your user ID, you are shown a Status for Table panel (Figure 164) with the value NOT OPEN in the STATUS FOR THIS SCREEN field.

----- STATUS FOR TABLE SAMTB1 -----

STATUS FOR THIS SCREEN : NOT OPEN	DATE CREATED : 85/06/15
TABLE AVAILABLE : YES	TIME CREATED : 10:15
	LAST DATE MODIFIED: 85/06/25
	LAST TIME MODIFIED: 14:52
	LAST MODIFIED BY : Z59SBL
	ORIGINAL ROW COUNT: 50
	CURRENT ROW COUNT : 60
	MODIFIED ROW COUNT: 15
	UPDATE COUNT : 4

Figure 164. Status for Table Panel with Table Not Open

The panel provides the following information:

STATUS FOR THIS SCREEN

Shows that the table is NOT OPEN for this logical screen.

TABLE AVAILABLE

YES or NO; whether you can open the table.

DATE CREATED

Date the table was created; shown in national format.

TIME CREATED

Time the table was created.

LAST DATE MODIFIED

Date the table was last modified; shown in national format.

LAST TIME MODIFIED

Time the table was last modified.

Dialog Test Tables (Option 7.4)

LAST MODIFIED BY

User ID of the user who last changed the table.

ORIGINAL ROW COUNT

The number of rows that were added to a newly created table before closing the table for the first time.

CURRENT ROW COUNT

The number of rows currently in the table.

MODIFIED ROW COUNT

The number of rows in the table that have been changed at least once. A row that has been added to an existing table is also considered a changed row.

UPDATE COUNT

Number of times the table has been modified. One or more updates during any table open/close sequence increments this counter by one.

The Modify Row option on the Tables panel allows you to change a key of a keyed table by adding the new row and deleting the old row. The row counts thus reflect this processing when changing a key value.

Table Open

If the table is open for your user ID, you are shown a Status for Table panel (Figure 165) with the value OPEN in the STATUS FOR THIS SCREEN field.

----- STATUS FOR TABLE SAMTB1 -----

STATUS FOR THIS SCREEN	: OPEN	DATE CREATED	: 85/06/15
OPEN OPTION	: WRITE	TIME CREATED	: 10:15
TABLE ON DISK	: YES	LAST DATE MODIFIED	: 85/06/25
LAST TABLE SERVICE	: TBSCAN	LAST TIME MODIFIED	: 14:52
LAST SERVICE RETURN CODE	: 0	LAST MODIFIED BY	: Z59SBL
CURRENT ROW POINTER	: 15	ORIGINAL ROW COUNT	: 50
		CURRENT ROW COUNT	: 60
		MODIFIED ROW COUNT	: 15
		UPDATE COUNT	: 4

Figure 165. Status for Table Panel with Table Open

This panel provides the following information:

STATUS FOR THIS SCREEN

Shows that the table is OPEN for this logical screen.

OPEN OPTION

Option used to open the table; this value can be WRITE, NOWRITE, SHR WRITE, or SHR NOWRITE.

TABLE ON DISK

Whether the table has been saved on disk; this value can be YES or NO.

LAST TABLE SERVICE

Name of the last table service invoked.

LAST SERVICE RETURN CODE

Last table services return code.

CURRENT ROW POINTER

Current position in the table.

DATE CREATED

Date the table was created; shown in national format.

TIME CREATED

Time the table was created.

LAST DATE MODIFIED

Date the table was last modified; shown in national format.

LAST TIME MODIFIED

Time the table was last modified.

LAST MODIFIED BY

User ID of the user who last changed the table.

ORIGINAL ROW COUNT

The number of rows that were added to a newly created table before closing the table for the first time.

CURRENT ROW COUNT

The number of rows currently in the table.

MODIFIED ROW COUNT

The number of rows in the table that have been changed at least once. A row that has been added to an existing table is also considered a changed row.

UPDATE COUNT

Number of times the table has been modified. One or more updates during any table open/close sequence increments this counter by one.

The Modify Row option on the Tables panel allows you to change a key of a keyed table by adding the new row and deleting the old row. The row counts thus reflect this processing when changing a key value.

Log (Option 7.5)

The Log option (7.5) allows you to display and browse data recorded in the ISPF log. You can use all the Browse commands, except BROWSE, while looking at the ISPF log.

ISPF Log Not Available

Sometimes the log is not available for browsing. This can occur when:

- The log data set is empty.
- The log data set was not created for this session because 0 was entered in the Primary pages field on the Log and List Defaults panel (option 0.2).
- No data has been written to the log during this session, and although the log data set exists and is not empty, you did not end the last ISPF session normally; for example, an ABEND may have ended the session. You can browse the log if you take an action that causes a log entry to be written.
- The log data set was previously allocated with a disposition of OLD. It must be allocated with a disposition of MOD.
- The log data set has been previously allocated to SYSOUT.

Trace Output in ISPF Log

The following trace output is written to the ISPF log:

- Trace header entries
- Function trace entries
- Variable trace entries.

Each type of entry follows the format of other log entries: a short summary on the left, and a detailed entry on the right.

Trace Header Entries

The first line of trace data is a trace header that identifies the trace and shows the current application ID, the current function, and the current screen. For split-screen mode, the original screen is 1 and the screen generated by the SPLIT command is 2. The summary section of the header entry identifies the entry as a dialog trace. The trace header entry is written during the test session whenever a function or variable trace entry is to be written for an application, function, or screen that differs from the last.

For example, a trace of logical screen 1 of function TESTF1 in application ISR would place the following line in the ISPF log:

Function Trace Entries

A pair of function trace entries, a BEGIN entry and an END entry, is generated during a function trace for each traced dialog service that is invoked. A service can be called from a user dialog that is currently executing, or from a Dialog Test action for the user. The summary portion of each of these entries shows the name of the dialog service, whether it is the beginning or the end of its execution, and whether it was invoked indirectly from a Dialog Test panel. If the word TEST does not appear, the user's dialog invoked the service directly. For END entries, the service return code is shown on a second line. The detailed section of the log entries contains an image of the service call and the parameters used to invoke that service, using two lines if necessary. For example:

```
DISPLAY .. BEGIN ... TEST - DISPLAY PANEL(XYZ)
.
.
.
DISPLAY .. END... ... TEST - DISPLAY PANEL(XYZ)
..RETURN CODE (0)
```

There can be many log entries between the begin and end entries. For example, any active variable traces can cause log entries during a SELECT trace.

You should note the following about the service call image:

- The image is truncated after the second line.
- ISPEXEC calls are shown as coded in the dialog.
- ISPLINK and ISPLNK calls (except for the ISREDIT service) are displayed with their parameter values separated by commas. Name-lists are shown as coded in the dialog, in either string or structure format. Structure format includes the count, element length, and list of names. For a variable services parameter whose context is defined by the "name-list" parameter on the service call, the first four bytes of the parameter value are displayed in hexadecimal format (X'nnnnnnnn').
- Dialog Test calls are shown using the command call format without the ISPEXEC prefix.

Variable Trace Entries

Two variable trace entry lines are generated for each variable trace log entry. The variable can be referenced or set by a user dialog directly or indirectly by a dialog service, or explicitly set by a Dialog Test option for a user. The summary parts of these entry lines identify the trace. Line one shows the name of the variable, the pool that contains it (F for function, S for shared, P for profile), and an indicator (TEST) if a Dialog Test option set the value. Line two shows the operation done for the variable (GET, PUT, or CHG) and the name of the dialog service that did the operation for non-TEST entries.

The current value of the variable is printed in the detail section of the log entry and may span two lines. For example:

```
LIB1.... POOL(P) .... - VALUE(FLAG)
...GET by EDIT      -
```

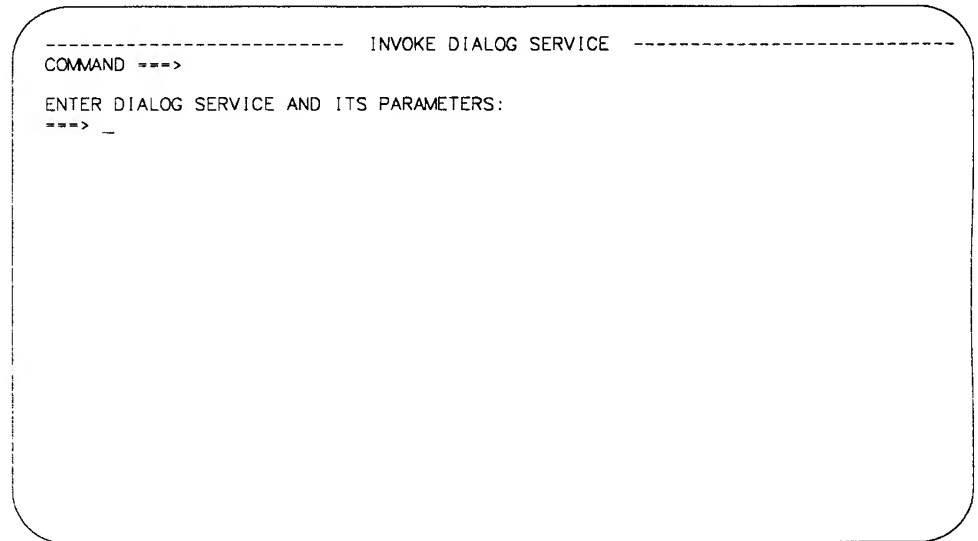
The value is truncated after the second line.

If the variable value contains characters that cannot be displayed, the value is displayed in hexadecimal format (X'nnnnnnnn').

Dialog Services (Option 7.6)

The Dialog Services option (7.6) allows you to invoke a dialog service by entering the service command invocation with or without the ISPEXEC characters.

Figure 166 on page 314 shows the Invoke Dialog Service panel.



```
----- INVOKE DIALOG SERVICE -----
COMMAND ==>
ENTER DIALOG SERVICE AND ITS PARAMETERS:
==> _
```

Figure 166. Invoke Dialog Service Panel

For example, if you want to display panel XYZ, enter:

```
COMMAND ==> DISPLAY PANEL XYZ
```

or:

```
COMMAND ==> ISPEXEC DISPLAY PANEL XYZ
```

The service is invoked when you press the ENTER key. You are informed of the service's completion and return code.

You can call any dialog service that is valid in the command environment except CONTROL at a breakpoint or before invoking a function.

If you issue the DISPLAY service call with only a message parameter, or with no parameter at all, the Special Display Panel is shown (Figure 167 on page 315).

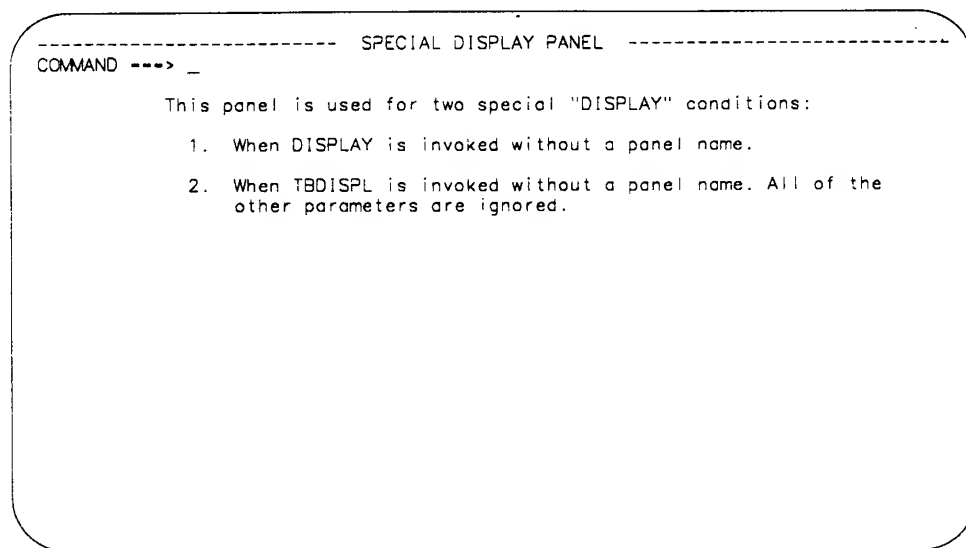


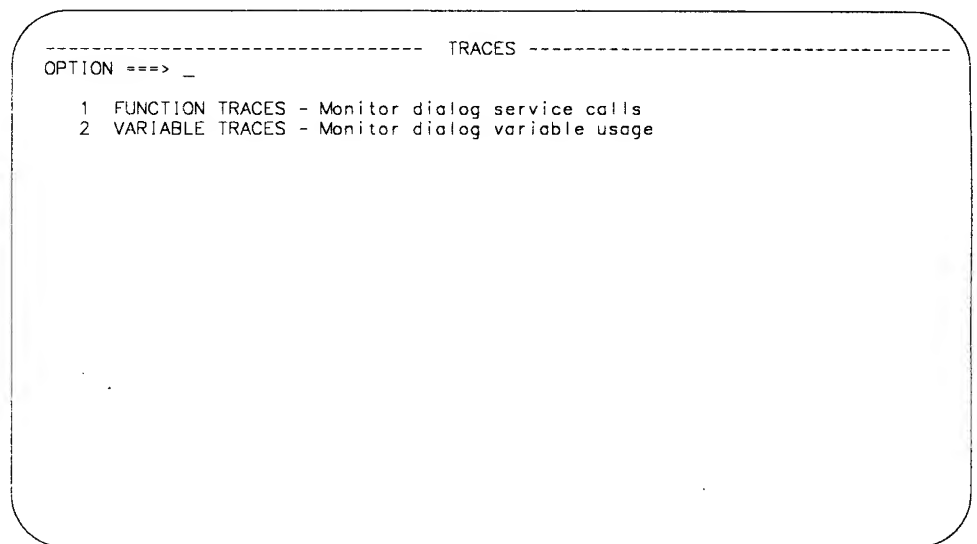
Figure 167. Special Display Panel

Traces (Option 7.7)

The Traces option (7.7) allows you to define, change, and delete trace specifications. You can trace executed dialog services, except for the VPUT or VGET service issued from a panel, and referenced dialog variables during dialog execution. Trace data is placed in the transaction log, where you can browse it by using the Log option (7.5), or print it when you leave ISPF. You can also print the log data set during a PDF session by using the ISPF LOG command.

Since tracing can degrade dialog performance and create large amounts of output, you should be careful in setting the scope of trace definitions.

When you select this option, a selection panel is displayed (Figure 168) on which you can show the type of trace you want to define.



```
----- TRACES -----
OPTION ===> _
1 FUNCTION TRACES - Monitor dialog service calls
2 VARIABLE TRACES - Monitor dialog variable usage
```

Figure 168. Traces Panel

The following sections describe the options shown at the top of the Traces panel.

1 - Function Traces

The Function Traces option on the Traces panel is used to establish criteria for recording the names of dialog service calls, the service parameters, and return code in the ISPF log. If either a dialog or Dialog Test processing causes a service call, that call is recorded in the trace. An example of Dialog Test processing that causes a service call is the use of the Panels option (7.2) to display a panel. Whenever a new application and/or function causes data to be recorded, a header is placed in the trace.

When you select the Function Traces option, you are shown a panel that you can scroll (Figure 169 on page 316). The panel lists all currently defined function traces.

Dialog Test - Traces (Option 7.7)

You can add, delete, and change function trace definitions by using this panel, either before invoking a function or at a breakpoint.

[illegible]

Figure 169. Function Traces Panel

Each line defines a function trace, showing:

''' Line command area.

FUNCTION The name of the user function that should contain the trace, or ALL to trace every dialog function. Initially, ALL is presented on the display but is not activated. Change the NO to a YES in the ACTIVE column to start such a trace. If you want to trace a function whose name is ALL, enclose the name in apostrophes to distinguish it; that is, specify 'ALL', not ALL.

ACTIVE Whether the trace is to be active now:

YES	The trace is currently active.
NO	The trace is currently not active.
Blank	The trace is currently active.

DIALOG SERVICES TO BE TRACED

Names of dialog services to be traced. No entry in this field shows all calls to dialog services for the function are to be traced.

All function traces exist until you leave Dialog Test, or until you delete them from this panel. Enter new information by typing over the existing data. The underscores are pad characters to show the starting and ending positions for each field; you do not need to blank them out. You can create several function traces before you press the ENTER key.

During dialog processing, to determine whether the criteria for a function trace have been met, Dialog Test executes a logical AND of the FUNCTION, ACTIVE, and DIALOG SERVICES fields specified for that

Function Traces Commands

2 - Variable Traces

When you select the Variable Traces option, you are shown a display that you can scroll (Figure 170). The display lists all currently defined variable traces. You can add, delete, and change variable trace definitions at a breakpoint, or by using this panel before invoking a function.

Figure 170. Variable Traces Panel

''' Line command area.

NW: 15457 DocId: 70002077 Page 683

Dialog Test - Traces (Option 7.7)

apostrophes to distinguish it; that is, specify 'ALL', not ALL.

POOL

Pool of interest for variable tracing:

F	Function variable pool.
S	Shared variable pool.
P	Profile variable pool.
Blank	All pools.

OPERATION

Type of variable reference to trace:

GET	Accesses to the variable's value.
PUT	Stores to the variable's value.
CHG	Changes to the variable's value.
Blank	All references to variable are traced.

FUNCTION

The user dialog function for which the trace is defined. If no entry, shows this variable is traced for all functions.

ACTIVE

Indication of whether the trace is to be active:

YES	The trace is currently active.
NO	The trace is currently not active.
Blank	The trace is currently active.

All variable trace definitions exist until you leave Dialog Test, or until you delete them from this panel. Enter new information by typing over the existing data. The underscores are pad characters to show the start and end of each field; you do not need to blank them out. You can create several variable traces before you press the ENTER key.

During dialog processing, to determine whether the criteria for a variable trace have been met, Dialog Test executes a logical AND of the VARIABLE, POOL, OPERATION, FUNCTION, and ACTIVE fields specified for that variable trace. Therefore, if you want more than one trace for a variable, you should create multiple rows.

Variable Traces Commands

The Variable Traces option uses the CANCEL, END, and LOCATE commands, and the D (delete), I (insert), and R (repeat) Dialog Test line commands. "Commands" on page 282 describes these commands.

Dialog Test - Breakpoints (Option 7.8)

WHEN	Indication of when the breakpoint should occur:
BEFORE	Before the service receives control.
AFTER	After the service completes execution.
Rnn	After the service completes execution, but only if the return code is the integer <i>nn</i> .
Blank	Before and after service execution.
FUNCTION	The program function or command function that must be executing for the breakpoint to be taken. No entry in this field shows that the breakpoint can occur for all functions.
ACTIVE	Indication of whether the breakpoint is to be active now:
YES	It is currently active.
NO	It is currently not active.
Blank	It is currently active.
QUAL	If present at the end of a row, shows that qualification data exists for the breakpoint. This field is non-modifiable. See "Qualification Parameter Values" for additional information.

All input fields contain underscores. Empty lines are added to the first display to fill up the screen. If you delete all the lines used for defining breakpoints, the display is automatically refreshed with enough empty lines to fill the screen again.

All breakpoints exist until you end or cancel your Dialog Test session, or until you delete them from this panel. Enter new information by typing over the existing data. The underscores are pad characters to show the starting and ending positions for each field; you do not need to blank them out. You can create several breakpoints before you press the ENTER key.

Breakpoints Commands

From the Breakpoints panel, you can use the CANCEL, END, LOCATE, QUAL, and RESUME commands, and the D (delete), I (insert), and R (repeat) Dialog Test line commands. "Commands" on page 282 describes these commands.

Qualification Parameter Values

A different part of the Breakpoints panel allows you to further constrain the conditions under which a breakpoint is to occur by letting you enter qualification parameter values. On this part of the panel, you can list parameter data with which the named service must have been invoked.

The Breakpoints panel with qualification parameter values is displayed (Figure 172 on page 321) when you enter the QUAL primary command on the first part of the Breakpoints panel. The FUNCTION and ACTIVE

[illegible][illegible]

!

100

- !

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
84

- Figure 1**

[illegible]

For example, if a SELECT call is SELECT PGM(ABC) PARM(1 2 3 5 '6'), then all or any of the following strings can be used: SELECT, PGM, ABC, 1, 2, 3, 5, 6.

For a breakpoint to be taken, all qualification data listed must be matched.

All line commands and change capabilities are still available on the Breakpoints panel with qualification parameter values.

During dialog processing, to determine whether the criteria for a breakpoint have been met, Dialog Test executes a logical AND of the SERVICE, WHEN, FUNCTION, ACTIVE, and QUALIFICATION fields specified for that breakpoint. Therefore, if you want more than one breakpoint for an ISPF service, you should create multiple rows.

When you use the Breakpoints option (7.8), be aware of the following items:

Qualification

If you plan to qualify several breakpoints, it may be more efficient to specify all breakpoint data on the Breakpoints panel with qualification parameter values.

END command

You can use the END primary command from either the first Breakpoints panel or the Breakpoints panel with qualification parameter values.

Input errors

You must correct input errors before leaving any display via the END, QUAL, or RESUME command. You can use the CANCEL command to end the Breakpoints option, even if input errors remain on the display.

Syntax checking

A dialog service call must pass a basic syntax check before a breakpoint is honored.

Control display

If any CONTROL service settings for DISPLAY LINE or DISPLAY SM (Session Manager) were in effect before the breakpoint, such settings are lost.

Finding a Breakpoint

If you invoke a dialog function or selection panel and find a breakpoint, the Breakpoint Primary Option Menu is displayed. Figure 173 shows this selection panel at a breakpoint just after the ISPF DISPLAY service was invoked while executing the TEST function in application PAY.

```

BREAKPOINT PRIMARY OPTION MENU - AFTER DISPLAY -----
OPTION ----> _

1  FUNCTIONS      - Invoke dialog function/selection menus
2  PANELS         - Display panels
3  VARIABLES      - Display/set variable information
4  TABLES        - Display/modify table information
5  LOG            - Browse ISPF log
6  DIALOG SERVICES - Invoke dialog services
7  TRACES         - Specify trace definitions
8  BREAKPOINTS    - Specify breakpoint definitions
T  TUTORIAL       - Display information about Dialog Test
G  GO             - Continue execution from breakpoint
C  CANCEL         - Cancel dialog test

CURRENT STATUS:
APPLICATION: PAY          FUNCTION: TEST
BREAKPOINT : ISPEXEC     DISPLAY  PANEL(TESTP) MSG()

RETURN CODE ----> 0
    
```

Figure 173. Breakpoint Primary Option Menu

Like the Dialog Test Primary Option Menu, the Breakpoint Primary Option Menu allows you to use the RETURN command from any one of the selected test options to display the Breakpoint Primary Option Menu again. At the Breakpoint Primary Option Menu, the END and RETURN commands have no effect. You must use the Go option (G) to end processing at this breakpoint and continue executing the dialog being tested, or the Cancel option (C) to cancel the Dialog Test option (7). This protects against inadvertent loss of data.

The Breakpoint Primary Option Menu contains all the options of the Dialog Test Primary Option Menu except Exit (7.X) and, as such, presents all but one of the Dialog Test functions to you.

This panel also contains two options not shown on the Dialog Test Primary Option Menu: Go (G) and Cancel (C). When a breakpoint occurs, these options let you continue execution or stop execution, respectively:

G GO The Go option continues dialog execution from a breakpoint. The user dialog resumes processing from the point at which it was suspended.

C CANCEL The Cancel option ends dialog testing and displays the first primary option panel you displayed at the beginning of your ISPF session again. All trace and breakpoint definitions are lost when Dialog Test is terminated.

When a user dialog finds a breakpoint, the current dialog environment is saved. When you select the Go option, the environment is restored, except for the following:

- If you change variable, table, and file tailoring data at a breakpoint, these actions are an extension of the suspended dialog; it is as though the dialog had taken all the actions itself during execution.
- If you change the service return code on the Breakpoint Primary Option Menu, the new return code is passed back to the dialog as though the service had set the new code itself.
- If you execute the PANELID command at the breakpoint, the last setting for displaying panel identifiers is retained.
- If any CONTROL service settings for DISPLAY LINE or DISPLAY SM (Session Manager) were in effect before the breakpoint, such settings are lost.

Note that the manipulation of one dialog part can cause a change to another dialog part. For example, if a panel is displayed, variables can be set.

All trace and breakpoint definitions are lost if you select the Cancel option.

The Breakpoint Primary Option Menu also displays the following information:

AFTER or BEFORE

An indication of whether the dialog has been suspended BEFORE or AFTER the service has executed.

Service Name

The name of the service at which the dialog has been suspended. In Figure 173 on page 323, the service name is DISPLAY.

CURRENT STATUS:

The application's current status when the breakpoint occurred. The following fields show this status:

APPLICATION

The application identifier of the suspended user dialog.

FUNCTION

The program or command name of the suspended user dialog.

BREAKPOINT

Up to three display lines showing an image of the dialog service call. Truncation occurs after the second line.

ISPEXEC calls are shown as coded.

ISPLINK (ISPLNK) calls are displayed with their parameter values separated by commas. Name-lists are

shown as coded in the dialog, in string format or in structure format. Structure format includes the count, element length, and list of names. For variable services parameters whose context is defined by the "name-list" parameter on the service call (for example, the variable value areas for a VDEFINE), the first four bytes of the parameter value are displayed in hexadecimal format (X'nnnnnnnn').

ISPEXEC calls from a program are the same as ISPEXEC calls from a command except that ISPEXEC is not displayed.

RETURN CODE

The dialog service return code. This field is displayed only if the breakpoint occurs *after* the dialog service has executed. The RETURN CODE field is modifiable; its value is passed back to the dialog (as the service's) when you select the Go option. This helps test dialog error handling.

Dialog Test - Tutorial (Option 7.T) and Exit (Option 7.X)

Tutorial (Option 7.T)

The Tutorial option (7.T) allows you to display information about the Dialog Test facilities. Figure 174 shows the first panel displayed when you select the Tutorial option.

```
----- DIALOG TEST TUTORIAL -----
OPTION ----> _

                                ISPF/PDF DIALOG TEST
                                TUTORIAL

This tutorial provides on-line information about the features and operation
of Dialog Test.

The Dialog Test tutorial consists of two parts: one describes the Dialog
Test option, as selected from the ISPF/PDF Primary Option Menu, and the
other describes the Dialog Test facilities available when a user dialog
encounters a "breakpoint" in its processing.

Beginning users are advised to review the Dialog Test Option topic first.

The following topics are presented in sequence, or may be selected by number:

    1 - Dialog Test Option
    2 - At A User Dialog Breakpoint
PF 1=HELP    2=SPLIT    3=END      4=RETURN    5=RFIND    6=RCHANGE
PF 7=UP      8=DOWN     9=SWAP     10=LEFT     11=RIGHT   12=RETRIEVE
```

Figure 174. Dialog Test Tutorial - First Panel

The default PF key command assignments for a terminal with 12 PF keys are shown at the bottom of the screen if you enter the PFSHOW command. See "ISPF Commands Assigned to PF Keys" on page 11 for the definitions of these commands.

Exit (Option 7.X)

The Exit option (7.X) ends your Dialog Test session. All trace and breakpoint definitions are lost.

CHAPTER 11. LIBRARY MANAGEMENT UTILITIES (Option 8)

This chapter provides an introduction to the Library Management Utilities, option 8 on the ISPF/PDF Primary Option Menu. This option allows you to use Library Management Facility (LMF) functions to:

- Effectively maintain control over different versions of the same application program
- Reduce the risks of writing over members accidentally and of two or more users modifying the same member simultaneously.

The chapter supplies information for project administrators and shows you the Library Management Utilities panel. For more information about LMF, see *Library Management*.

Chapter 11. LM Utilities (Option 8)

When you select the Library Management Utilities option (8) from the ISPF/PDF Primary Option Menu, the panel shown in Figure 175 is displayed. The options on this panel give you access to the Library Management Facility (LMF), which is an extension of the ISPF library concept.

Notes:

1. *To use the Library Management Utilities option, you must have preallocated the ISRCFIL control file.*
2. *For complete information on the use of the Library Management Utilities option, see Library Management.*

```

----- LIBRARY MANAGEMENT UTILITIES -----
OPTION ==> _

1  CONTROLS      - Specify library attributes:
                   - Create, update, or delete promotion controls.
                   - Predefine, update or delete member controls.
2  USER SET      - Create, update, or delete list of users authorized to
                   - perform library functions.
3  DISTRIBUTION   - Create, update, or delete a distribution table to relate
                   - different promotion hierarchies.
4  ACTIVATE       - Activate or deactivate promotion hierarchy controls.
5  REVIEW         - Browse or print activity and library controls
                   - information.
6  ACTIVITY LOG   - Browse or remove entries from activity logs.
7  DS FULL ACTION - Specify actions to take when LMF data sets fill up.
T  TUTORIAL       - General information about the library management
                   - utility facility.

```

Figure 175. Library Management Utilities Panel

LMF allows you to control the contents of your libraries and manage your development processes. Therefore, libraries that you control by using LMF are called *controlled libraries*.

LMF requires a library administrator to set up and maintain library controls. If you create the controls for a library for a new project, you automatically become the project administrator for that project.

The controls you establish allow authorized users to:

- Enter information into a controlled library.

LMF, with help from RACF or an equivalent protection system, helps you maintain data integrity by allowing only one person at a time to update a member in a controlled library.

- Get information from a controlled library to change it and return it to the controlled library.

LMF allows only users who have been authorized by the project administrator to replace a member in a controlled library with an updated member.

- Move information from one controlled library to another.

LMF allows authorized users to put a member into one or more libraries, or one or more hierarchies within the same project.

The project administrator is the only person who can update the library controls and execute the User Set (option 8.2), Distribution (option 8.3), Activate (option 8.4), Activity Log (options 8.6.3 and 8.6.4) and DS Full Action (option 8.7) utilities for that project.

Only the project administrator can change the project administrator ID of the project. Where there are several project administrators for a given project, only one administrator can update the project controls at a time. The updating of project controls is restricted to one logical screen.

You can add a row with no values to the table, but you are asked to confirm such an action to guard against inadvertent use of the END command.

When using the Add Row option, be aware of the following points:

- The position of the new row in the table depends on whether the table was previously sorted using the TBSORT dialog service. If the table was sorted, the new row is placed in sort order; if it has not been sorted, the new row is placed after the row you specified.
- You cannot delete a key or name variable and its value from the display or table row.
- You cannot change the variable name for a key or name variable; if you do, an error message is displayed and the original name is restored.
- If more than one variable entry has the same name, all instances of that variable are assigned the value of the last occurrence of the variable; that is, the occurrence closest to the bottom of the display that you can scroll.
- If the table has keys, the values for the keys in the added row must be different from those in all the existing rows when you leave the Add Row option. Otherwise, a message is displayed and the row is displayed again so you can change the keys.
- Blank save names are ignored and do not need to be deleted, even if data is left in the value.
- Hexadecimal data that usually cannot be displayed is translated to displayable characters or entered using the form:

X'nnnnnnnn'

where:

- | | |
|---|---|
| n | An integer 0 through 9 or an alphabetic character A through F. There must be an even number of characters within the quotation marks. |
|---|---|
- Variables defined with the VDEFINE service as non-character are shown in translated form; do not change them by using the hexadecimal representation. A format or length error causes an error message to be displayed when you use the END command.

Add Row Commands

The Add Row option uses the CANCEL, END, and LOCATE commands, and the D (delete), I (insert), and R (repeat) Dialog Test line commands. "Commands" on page 282 describes these commands. Inserted and repeated lines always have a type of S, because you can add only save variables to a row of an existing table. Also, you can delete only save variables.

6 - Display Status

If you select the Display Status option from the Tables panel, one of two data information panels is displayed for the table specified in the TABLE NAME field. The information reflects all operations using the specified table, including those done at your request by the Tables options under Dialog Test.

Table Not Open

If the table is not open for your user ID, you are shown a Status for Table panel (Figure 164) with the value NOT OPEN in the STATUS FOR THIS SCREEN field.

----- STATUS FOR TABLE SAMTB1 -----

STATUS FOR THIS SCREEN : NOT OPEN	DATE CREATED : 85/06/15
TABLE AVAILABLE : YES	TIME CREATED : 10:15
	LAST DATE MODIFIED: 85/06/25
	LAST TIME MODIFIED: 14:52
	LAST MODIFIED BY : Z59SBL
	ORIGINAL ROW COUNT: 50
	CURRENT ROW COUNT : 60
	MODIFIED ROW COUNT: 15
	UPDATE COUNT : 4

Figure 164. Status for Table Panel with Table Not Open

The panel provides the following information:

STATUS FOR THIS SCREEN

Shows that the table is NOT OPEN for this logical screen.

TABLE AVAILABLE

YES or NO; whether you can open the table.

DATE CREATED

Date the table was created; shown in national format.

TIME CREATED

Time the table was created.

LAST DATE MODIFIED

Date the table was last modified; shown in national format.

LAST TIME MODIFIED

Time the table was last modified.

CSP/AD (Option 9.1)

If the Cross System Product/Application Development (CSP/AD) licensed program is not installed or available on your system, PDF displays the panel shown in Figure 177:

```
----- ADDITIONAL IBM PROGRAM DEVELOPMENT PRODUCTS -----
COMMAND ----> _

| Cross System Product/Application Development |
|                               not installed                               |
|-----|

CSP/AD is not currently available to your terminal session. The initial
CSP/AD panel does not exist in your panel library. CSP/AD is not part of
ISPF/PDF, but may be installed along with ISPF/PDF.

CSP/AD is used to develop high quality end user applications which can
operate in most of the IBM operating environments for the 8100, 4300 and
30xx processors. CSP/AD has demonstrated a significant increase in
productivity over that realized by more traditional methods.

For more information on CSP you may contact IBM Marketing or order these
publications:
  GH23-0500 : CSP/AD and CSP/AE General Information
  SH23-0501 : CSP/AD User's Guide
  SH23-0502 : CSP/AD Operation - Development
  GX23-0900 : CSP/AD Reference Summary
```

Figure 177. Cross System Product/Application Development Panel

OPEN OPTION

Option used to open the table; this value can be WRITE, NOWRITE, SHR WRITE, or SHR NOWRITE.

TABLE ON DISK

Whether the table has been saved on disk; this value can be YES or NO.

LAST TABLE SERVICE

Name of the last table service invoked.

LAST SERVICE RETURN CODE

Last table services return code.

CURRENT ROW POINTER

Current position in the table.

DATE CREATED

Date the table was created; shown in national format.

TIME CREATED

Time the table was created.

LAST DATE MODIFIED

Date the table was last modified; shown in national format.

LAST TIME MODIFIED

Time the table was last modified.

LAST MODIFIED BY

User ID of the user who last changed the table.

ORIGINAL ROW COUNT

The number of rows that were added to a newly created table before closing the table for the first time.

CURRENT ROW COUNT

The number of rows currently in the table.

MODIFIED ROW COUNT

The number of rows in the table that have been changed at least once. A row that has been added to an existing table is also considered a changed row.

UPDATE COUNT

Number of times the table has been modified. One or more updates during any table open/close sequence increments this counter by one.

The Modify Row option on the Tables panel allows you to change a key of a keyed table by adding the new row and deleting the old row. The row counts thus reflect this processing when changing a key value.

INFO/SYS (Option 9.3)

If the Information/System (INFO/SYS) product is not installed or available on your system, PDF displays the panel shown in Figure 179:

```
----- ADDITIONAL IBM PROGRAM DEVELOPMENT PRODUCTS -----  
COMMAND ===> _  
  
      | Information/System is not available |  
      |-----|  
  
Information/System is not currently available to your terminal session.  
The initial Information/System panel does not exist in your panel library.  
Information/System is not part of ISPF/PDF, but may be installed along with  
ISPF/PDF.  
  
IBM's Information/Family of products provides you with an organized  
framework which helps you to manage your data processing problems and  
changes and to maintain your installation's inventory and configuration in  
an orderly way. It aids in the management of both daily control and  
planning of your installation's growth.  
  
For more information on Information/Family you may contact IBM Marketing or  
order these publications:  
  GC34-4045 : Introducing the Information/Family  
  SC34-4046 : Planning and Installing the Information/Family  
  SC34-4052 : Using the Information/Family Licensed Programs
```

Figure 179. Information/System Panel

Function Trace Entries

A pair of function trace entries, a BEGIN entry and an END entry, is generated during a function trace for each traced dialog service that is invoked. A service can be called from a user dialog that is currently executing, or from a Dialog Test action for the user. The summary portion of each of these entries shows the name of the dialog service, whether it is the beginning or the end of its execution, and whether it was invoked indirectly from a Dialog Test panel. If the word TEST does not appear, the user's dialog invoked the service directly. For END entries, the service return code is shown on a second line. The detailed section of the log entries contains an image of the service call and the parameters used to invoke that service, using two lines if necessary. For example:

```
DISPLAY .. BEGIN ... TEST - DISPLAY PANEL(XYZ)
.
.
.
DISPLAY .. END... ... TEST - DISPLAY PANEL(XYZ)
..RETURN CODE (0)
```

There can be many log entries between the begin and end entries. For example, any active variable traces can cause log entries during a SELECT trace.

You should note the following about the service call image:

- The image is truncated after the second line.
- ISPEXEC calls are shown as coded in the dialog.
- ISPLINK and ISPLNK calls (except for the ISREDIT service) are displayed with their parameter values separated by commas. Name-lists are shown as coded in the dialog, in either string or structure format. Structure format includes the count, element length, and list of names. For a variable services parameter whose context is defined by the "name-list" parameter on the service call, the first four bytes of the parameter value are displayed in hexadecimal format (X'nnnnnnnn').
- Dialog Test calls are shown using the command call format without the ISPEXEC prefix.

Variable Trace Entries

Two variable trace entry lines are generated for each variable trace log entry. The variable can be referenced or set by a user dialog directly or indirectly by a dialog service, or explicitly set by a Dialog Test option for a user. The summary parts of these entry lines identify the trace. Line one shows the name of the variable, the pool that contains it (F for function, S for shared, P for profile), and an indicator (TEST) if a Dialog Test option set the value. Line two shows the operation done for the variable (GET, PUT, or CHG) and the name of the dialog service that did the operation for non-TEST entries.

COBOL/SF-B (Option 9.5)

If the COBOL Structuring Facility (COBOL/SF) is not installed or available on your system, PDF displays the panel shown in Figure 181:

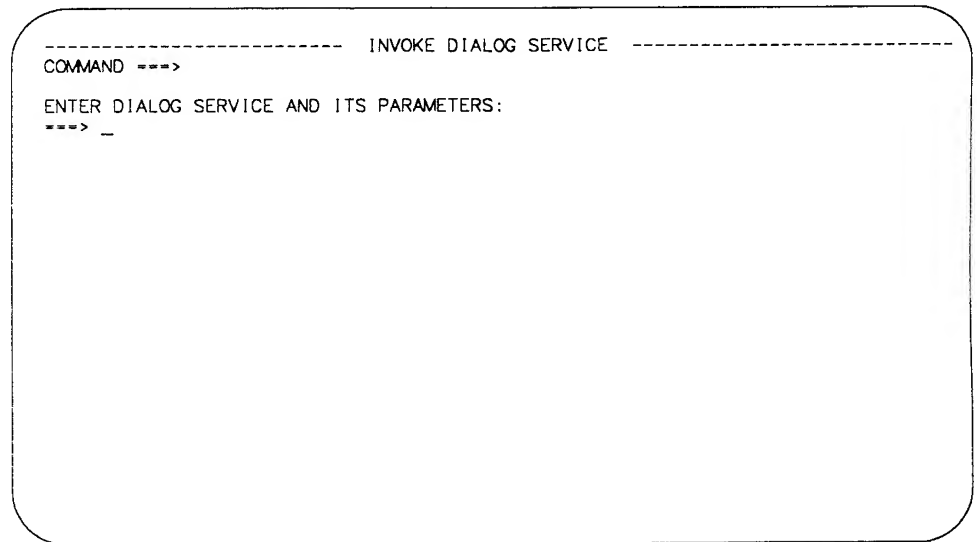
```
----- ADDITIONAL IBM PROGRAM DEVELOPMENT PRODUCTS -----  
COMMAND ----> _  
  
      |-----|  
      | COBOL Structuring Facility |  
      | is not installed           |  
      |-----|  
  
COBOL Structuring Facility (COBOL/SF) is not installed on your system or is  
not available on your terminal session. COBOL/SF is not part of  
ISPF/PDF, but may be installed along with ISPF/PDF.  
  
COBOL/SF (5668-786) transforms unstructured VS COBOL II programs into  
structured VS COBOL II for improved understandability and maintainability.  
It also produces an extensive Re-engineering Report of the structural  
qualities of the input program and the output program.  
  
For more information on COBOL/SF you may contact IBM Marketing or order  
these publications:  
  
GC34-4078 : COBOL Structuring Facility Licensed Program Specifications  
SC34-4079 : COBOL Structuring Facility Re-engineering Concepts  
SC34-4080 : COBOL Structuring Facility User's Guide and Reference
```

Figure 181. COBOL Structuring Facility Panel - Background Dialog

Dialog Services (Option 7.6)

The Dialog Services option (7.6) allows you to invoke a dialog service by entering the service command invocation with or without the ISPEXEC characters.

Figure 166 on page 314 shows the Invoke Dialog Service panel.



```
----- INVOKE DIALOG SERVICE -----
COMMAND ===>
ENTER DIALOG SERVICE AND ITS PARAMETERS:
===> _
```

Figure 166. Invoke Dialog Service Panel

For example, if you want to display panel XYZ, enter:

```
COMMAND ===> DISPLAY PANEL(XYZ)
```

or:

```
COMMAND ===> ISPEXEC DISPLAY PANEL(XYZ)
```

The service is invoked when you press the ENTER key. You are informed of the service's completion and return code.

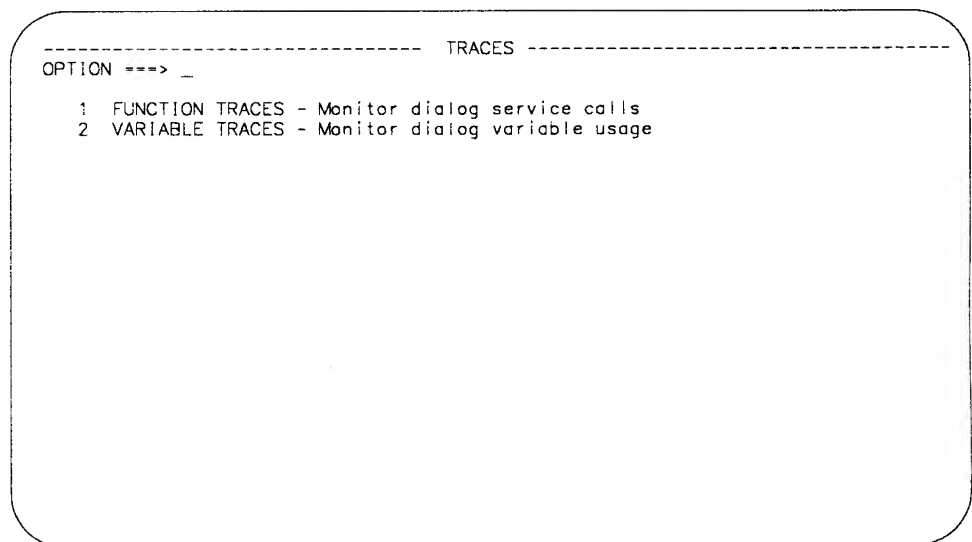
You can call any dialog service that is valid in the command environment except CONTROL at a breakpoint or before invoking a function.

Traces (Option 7.7)

The Traces option (7.7) allows you to define, change, and delete trace specifications. You can trace executed dialog services, except for the VPUT or VGET service issued from a panel, and referenced dialog variables during dialog execution. Trace data is placed in the transaction log, where you can browse it by using the Log option (7.5), or print it when you leave ISPF. You can also print the log data set during a PDF session by using the ISPF LOG command.

Since tracing can degrade dialog performance and create large amounts of output, you should be careful in setting the scope of trace definitions.

When you select this option, a selection panel is displayed (Figure 168) on which you can show the type of trace you want to define.



```
----- TRACES -----
OPTION ----> _
1 FUNCTION TRACES - Monitor dialog service calls
2 VARIABLE TRACES - Monitor dialog variable usage
```

Figure 168. Traces Panel

The following sections describe the options shown at the top of the Traces panel.

1 - Function Traces

The Function Traces option on the Traces panel is used to establish criteria for recording the names of dialog service calls, the service parameters, and return code in the ISPF log. If either a dialog or Dialog Test processing causes a service call, that call is recorded in the trace. An example of Dialog Test processing that causes a service call is the use of the Panels option (7.2) to display a panel. Whenever a new application and/or function causes data to be recorded, a header is placed in the trace.

When you select the Function Traces option, you are shown a panel that you can scroll (Figure 169 on page 316). The panel lists all currently defined function traces.

Batch Processing Sequence

You will use these data sets later if you need to correct your program using COBOL interactive debug (option 4.10A).

Note: You do not have to preallocate symbolic debug and print output data sets to use VS COBOL II interactive debug (options 4.10 and 5.10).

Allocate these data sets and then continue with the next step. See “Symbolic Debug Data Sets” on page 242 and “Print Output Data Sets” on page 243 if you need more information.

- If you have compiled a program using the OS/VS COBOL compiler and now want to run COBOL interactive debug, you must first use the linkage editor (option 4.7 or option 5.7) to generate a load module.

When you have satisfied these requirements or chosen a different processor, continue with the next step.

2. If you do not know whether the source data is in packed format, find out by editing the data set and entering the PROFILE command. The profile shows either PACK ON (data is stored in packed format) or PACK OFF (data is not stored in packed format).

If the data is not packed, continue with the next step.

If the data is packed, you should read “Expanding Packed Data” on page 211, paying close attention to information that applies to the batch processor you plan to use. When you are satisfied that the data set is ready to be processed, save the data set if you are in Edit and continue with the next step.

3. Select a batch processor. If you bypass the Batch Selection Panel, you cannot verify or change the job statement parameters, or generate multiple compilations (multiple job steps) or link edits within the same job.
4. Enter YES or NO in the SOURCE DATA ONLINE field to tell PDF whether the data to be processed resides on a currently mounted volume. If you specify YES, PDF checks the data set information that you entered on the Batch Selection Panel, and allows you to display a member list. If you specify NO, PDF assumes that the data cannot be accessed except by the batch job, and does not verify the existence or validity of the specified data set.
5. Enter YES or NO in the SOURCE DATA PACKED field to tell PDF whether it will need to expand the source data.

Note: The SOURCE DATA PACKED field has no effect on the member parts list option (5.12). Member parts list can read both packed and unpacked data sets, so no expansion is needed.

6. Enter any job statement information you need. See “Job Statement Information” on page 55 if you need more information.

Function Traces Commands

2 - Variable Traces

When you select the Variable Traces option, you are shown a display that you can scroll (Figure 170). The display lists all currently defined variable traces. You can add, delete, and change variable trace definitions at a breakpoint, or by using this panel before invoking a function.

Figure 170. Variable Traces Panel

''' Line command area.

NW: 15457 DocId: 70002077 Page 706

Batch Processing Sequence

If you need information about the options or about other fields on the panel, go to the page referred to in the following list. Then continue with step 12.

Processor	Reference
Assembler H	"Assembler H (Option 5.1)" on page 258
Assembler XF	"Assembler XF (Option 5.1A)" on page 260
VS COBOL II compiler	"VS COBOL II Compiler (Option 5.2)" on page 262
OS/VS COBOL compiler	"OS/VS COBOL Compiler (Option 5.2A)" on page 264
VS FORTRAN compiler	"VS FORTRAN Compiler (Option 5.3)" on page 265
PL/I checkout compiler	"PL/I Checkout Compiler (Option 5.4)" on page 267
PL/I optimizing compiler	"PL/I Optimizing Compiler (Option 5.5)" on page 268
Pascal/VS compiler	"Pascal/VS Compiler (Option 5.6)" on page 269
Linkage editor	"Linkage Editor (Option 5.7)" on page 270
VS COBOL II interactive debug	"VS COBOL II Interactive Debug (Option 5.10)" on page 272

12. Enter any additional input libraries you need. For VS COBOL II interactive debug, enter any input LOAD libraries that you need to complete the search. These libraries must be LOAD libraries only. See "Input Data Sets" on page 215 if you need help.

13. Once all the input fields have been specified, press the ENTER key to invoke the batch processor. ISPF generates the appropriate JCL statements. See "JCL Generation - Compilers" on page 256 and "JCL Generation - Assemblers and Linkage Editor" on page 257 if you need more information.

Note: You can leave the entry panel without generating any JCL by entering the END command instead of pressing the ENTER key.

14. One of the following occurs:

- If you used the jump function to bypass the Batch Selection Panel, PDF submits the generated JCL and returns directly to the ISPF/PDF Primary Option Menu.

JCL Generation - Compilers

Figure 144 on page 268 shows an example for the PL/I optimizing compiler. This panel is typical of the batch compiler entry panels. After you fill in an entry panel and press the ENTER key, PDF generates the appropriate JCL statements. The JCL that would be generated for the PL/I example is:

```
//SCAN      EXEC  PGM=ISRSCAN,PARM='TOPSEG',COND=(12,LE)
//IN         DD   DSN=ISPFDEMO.MYLIB.PLI,DISP=SHR
//          DD   DSN=ISPFDEMO.MASTER.PLI,DISP=SHR
//          DD   DSN=ISPFDEMO.FLAG.PLI,DISP=SHR
//OUT        DD   UNIT=SYSDA,DISP=(NEW,PASS),SPACE=(CYL,(2,2)),
//          DSN=&&TEMP1
//*
//PLIO       EXEC  PGM=IELOAA,REGION=100K,COND=(12,LE),
//          PARM='MACRO,XREF'
//SYSPRINT   DD   SYSOUT=(A)
//SYSIN      DD   DSN=&&TEMP1,DISP=(OLD,DELETE)
//SYSUT1     DD   UNIT=SYSDA,SPACE=(CYL,(2,2))
//SYSLIB     DD   DSN=ISPFDEMO.MYLIB.PLI,DISP=SHR
//          DD   DSN=ISPFDEMO.MASTER.PLI,DISP=SHR
//          DD   DSN=ISPFDEMO.FLAG.PLI,DISP=SHR
//          DD   DSN=OURSYS.PLIMACS,DISP=SHR
//SYSLIN     DD   DSN=ISPFDEMO.MYLIB.OBJ(TOPSEG),DISP=OLD
```

The JCL is generated in two steps:

1. The first step executes one of the following scan programs, which are distributed as part of PDF:

ISRSCAN Copies one member.

ISRLEMX Copies the primary member, expands any included members, and unpacks any packed members.

The selected scan program searches the user-specified sequence of concatenated libraries to find the designated member. If the scan program finds the member, it copies the member to a temporary sequential data set that is indicated by &&TEMP1 and generated by the system. The scan program then exits with a return code of zero, if no errors are found. If any errors are found, the scan program exits with one of the following return codes, which prevent execution of the second job step.

columns are overlaid with a column of data titled QUALIFICATION PARAMETER VALUES; this column was logically off the screen to the right of the first Breakpoints panel. To resume the format of the Breakpoints panel, use the RESUME primary command.

```

----- BREAKPOINTS ----- ROW 1 OF 13
COMMAND ----> _ SCROLL ----> PAGE

ADD, DELETE, AND CHANGE BREAKPOINTS.  UNDERSCORES NEED NOT BE BLANKED.
ENTER END COMMAND TO FINALIZE CHANGES.

      SERVICE      WHEN      QUALIFICATION PARAMETER VALUES
    (Required)    (BEFORE,AFTER,Rnn)    (No entry=none)
                          (No entry=all)    ("AND" is assumed between values)

****
**** _____
**** _____
**** _____
**** _____
**** _____
**** _____
**** _____
**** _____
**** _____
**** _____
**** _____
**** _____
**** _____
**** _____
**** _____
**** _____
***** BOTTOM OF DATA *****

```

Figure 172. Breakpoints Panel with Qualification Parameter Values

The lines on the Breakpoints panel with qualification parameter values correspond to the lines on the first Breakpoints panel; "Specifying Breakpoints" on page 319 describes the SERVICE and WHEN fields. In the QUALIFICATION PARAMETER VALUES field, you can enter any combination of the following for all services except SELECT:

- One or more parameter values, separated by blanks, that the dialog passes to the service. No order is implied by the specification of the parameter values.

For example, if you want a breakpoint to occur when message **ABC0001** is coded on a DISPLAY service request, you should specify ABC0001. If the breakpoint should occur only when message **ABC0001** and panel XYZ are both coded, you should specify ABC0001 XYZ.

- One or more command call keywords, separated by blanks, that have values that are not blank when a dialog calls the service. For ISPLINK or ISPLNK calls, the keywords matching the calling sequence parameter positions are used.

For example, if you want a breakpoint to occur whenever the DISPLAY service is invoked with a message, then specify MSG.

For the SELECT service, you can enter one or more parameter strings that would be entered on the SELECT service call. A string is a series of characters delimited by a blank, a comma, a single quotation mark, or a left or right parenthesis.

Batch - Assembler H (Option 5.1)

Assembler H (Option 5.1)

Batch Assembler H is invoked from the Batch Assembler H panel, shown in Figure 132. Assembler H does not use a prompter. For information about Assembler H allocation data sets, see Appendix D, “Allocation Data Sets” on page 369.

```
----- BATCH ASSEMBLER H -----
COMMAND ---->

ISPF LIBRARY:
PROJECT ----> ISPFDEMO
GROUP ----> XXX      ----> A      ----> MASTER  ---->
TYPE ----> ASM
MEMBER ----> TOP      (Blank or pattern for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ---->

LIST ID ----> LISTASM_      (Blank for hardcopy listing)
SYSOUT CLASS ---->      (If hardcopy requested)

ASSEMBLER OPTIONS:
TERM ----> NOTERM      (TERM or NOTERM)
OTHER ----> TEST,RENT

ADDITIONAL INPUT LIBRARIES:
----> 'ISPFTEST.FLAG.ASM'
---->
```

Figure 132. Batch Assembler H Panel

All the fields on this panel are explained in Chapter 2, “Libraries and Data Sets,” except LIST ID, which is explained in “List Data Sets” on page 216, ADDITIONAL INPUT LIBRARIES, which is explained in “Input Data Sets” on page 215, and the following:

TERM In the TERM field, enter TERM if you want PDF to generate a terminal data set. A *terminal data set* contains a synopsis of the error messages produced by Assembler XF. If the input data set is partitioned, the terminal data set name is:

prefix.userid.member.TERM

where “prefix” is your TSO prefix, if you have one and if it is different from your user ID, “userid” is your user ID, and “member” is the name of the member being assembled. However, if the input data set is sequential, the terminal data set name is:

prefix.userid.TEMPNAME.TERM

Enter NOTERM in the TERM field to avoid generating the terminal data set. If you leave this field blank, PDF assumes NOTERM.

OTHER Enter any other options you need in the OTHER field.

Finding a Breakpoint

If you invoke a dialog function or selection panel and find a breakpoint, the Breakpoint Primary Option Menu is displayed. Figure 173 shows this selection panel at a breakpoint just after the ISPF DISPLAY service was invoked while executing the TEST function in application PAY.

```
BREAKPOINT PRIMARY OPTION MENU - AFTER DISPLAY -----
OPTION ----> _

1  FUNCTIONS      - Invoke dialog function/selection menus
2  PANELS         - Display panels
3  VARIABLES      - Display/set variable information
4  TABLES        - Display/modify table information
5  LOG            - Browse ISPF log
6  DIALOG SERVICES - Invoke dialog services
7  TRACES         - Specify trace definitions
8  BREAKPOINTS    - Specify breakpoint definitions
T  TUTORIAL       - Display information about Dialog Test
G  GO             - Continue execution from breakpoint
C  CANCEL         - Cancel dialog test

CURRENT STATUS:
APPLICATION: PAY          FUNCTION: TEST
BREAKPOINT : ISPEXEC     DISPLAY  PANEL(TESTP)  MSG()

RETURN CODE ----> 0
```

Figure 173. Breakpoint Primary Option Menu

Like the Dialog Test Primary Option Menu, the Breakpoint Primary Option Menu allows you to use the RETURN command from any one of the selected test options to display the Breakpoint Primary Option Menu again. At the Breakpoint Primary Option Menu, the END and RETURN commands have no effect. You must use the Go option (G) to end processing at this breakpoint and continue executing the dialog being tested, or the Cancel option (C) to cancel the Dialog Test option (7). This protects against inadvertent loss of data.

The Breakpoint Primary Option Menu contains all the options of the Dialog Test Primary Option Menu except Exit (7.X) and, as such, presents all but one of the Dialog Test functions to you.

This panel also contains two options not shown on the Dialog Test Primary Option Menu: Go (G) and Cancel (C). When a breakpoint occurs, these options let you continue execution or stop execution, respectively:

G GO The Go option continues dialog execution from a breakpoint. The user dialog resumes processing from the point at which it was suspended.

C CANCEL

The Cancel option ends dialog testing and displays the first primary option panel you displayed at the beginning of your ISPF session again. All trace and breakpoint definitions are lost when Dialog Test is terminated.

Assembler XF (Option 5.1A)

Batch Assembler XF is invoked from the Batch Assembler XF panel, shown in Figure 134.

```
----- BATCH ASSEMBLER XF -----
COMMAND ---->

ISPF LIBRARY:
PROJECT ----> ISPFDEMO
GROUP  ----> XXX      ----> A      ----> MASTER  ---->
TYPE   ----> ASM
MEMBER ----> TOP      (Blank or pattern for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ---->

LIST ID  ----> LISTASM_      (Blank for hardcopy listing)
SYSOUT CLASS ---->      (If hardcopy requested)

ASSEMBLER OPTIONS:
TERM ---->      (TERM or NOTERM)
OTHER ----> LIST,RENT

ADDITIONAL INPUT LIBRARIES:
----> 'ISPFTEST.FLAG.ASM'
---->
```

Figure 134. Batch Assembler XF Panel

All the fields on this panel are explained in Chapter 2, “Libraries and Data Sets,” except LIST ID, which is explained in “List Data Sets” on page 216 , ADDITIONAL INPUT LIBRARIES, which is explained in “Input Data Sets” on page 215, and the following:

TERM In the TERM field, enter TERM if you want PDF to generate a terminal data set. A *terminal data set* contains a synopsis of the error messages produced by Assembler XF. If the input data set is partitioned, the terminal data set name is:

prefix.userid.member.TERM

where “prefix” is your TSO prefix, if you have one and it is different from your user ID, “userid” is your user ID, and “member” is the name of the member being assembled.

However, if the input data set is sequential, the terminal data set name is:

prefix.userid.TEMPNAME.TERM

Enter NOTERM in the TERM field to avoid generating the terminal data set. If you leave this field blank, PDF assumes NOTERM.

OTHER Enter any other options you need in the OTHER field.

shown as coded in the dialog, in string format or in structure format. Structure format includes the count, element length, and list of names. For variable services parameters whose context is defined by the "name-list" parameter on the service call (for example, the variable value areas for a VDEFINE), the first four bytes of the parameter value are displayed in hexadecimal format (X'nnnnnnnn').

ISPEXEC calls from a program are the same as ISPEXEC calls from a command except that ISPEXEC is not displayed.

RETURN CODE

The dialog service return code. This field is displayed only if the breakpoint occurs *after* the dialog service has executed. The RETURN CODE field is modifiable; its value is passed back to the dialog (as the service's) when you select the Go option. This helps test dialog error handling.

VS COBOL II Compiler (Option 5.2)

ISPF generates a COBOL command from the values you enter on the Batch VS COBOL II Compile panel, shown in Figure 136. VS COBOL II does not use a prompter. For information about VS COBOL II allocation data sets, see Appendix D, "Allocation Data Sets" on page 367.

```
----- BATCH VS COBOL II COMPILE -----
COMMAND ---->

ISPF LIBRARY:
PROJECT ----> ISPFDEMO
GROUP ----> XXX      ----> A      ----> MASTER      ---->
TYPE ----> COBOL
MEMBER ----> TOP      (Blank or pattern for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ---->

LIST ID ----> LSTCOBOL_ (Blank for hardcopy listing)
SYSOUT CLASS ----> (If hardcopy requested)

COMPILER OPTIONS:
TERM ----> (TERM or NOTERM)
OTHER ----> RESIDENT

ADDITIONAL INPUT LIBRARIES:
----> 'ISPFTEST.FLAG.COBOL'
---->
```

Figure 136. Batch VS COBOL II Compile Panel

All the fields on this panel are explained in Chapter 2, "Libraries and Data Sets," except LIST ID, which is explained in "List Data Sets" on page 216, ADDITIONAL INPUT LIBRARIES, which is explained in "Input Data Sets" on page 215, and the following:

TERM In the TERM field, enter TERM if you want PDF to generate a terminal data set. A *terminal data set* contains a synopsis of the error messages produced by Assembler XF. If the input data set is partitioned, the terminal data set name is:

prefix.userid.member.TERM

where "prefix" is your TSO prefix, if you have one and if it is different from your user ID, "userid" is your user ID, and "member" is the name of the member being assembled. However, if the input data set is sequential, the terminal data set name is:

prefix.userid.TEMPNAME.TERM

Enter NOTERM in the TERM field to avoid generating the terminal data set. If you leave this field blank, PDF assumes NOTERM.

OTHER If you plan to run VS COBOL II interactive debug after you compile your program, enter TEST, RESIDENT, and any other options you need in the OTHER field.

CHAPTER 11. LIBRARY MANAGEMENT UTILITIES (Option 8)

This chapter provides an introduction to the Library Management Utilities, option 8 on the ISPF/PDF Primary Option Menu. This option allows you to use Library Management Facility (LMF) functions to:

- Effectively maintain control over different versions of the same application program
- Reduce the risks of writing over members accidentally and of two or more users modifying the same member simultaneously.

The chapter supplies information for project administrators and shows you the Library Management Utilities panel. For more information about LMF, see *Library Management*.

OS/VS COBOL Compiler (Option 5.2A)

```

----- BATCH OS/VS COBOL COMPILE -----
COMMAND ---->

ISPF LIBRARY:
PROJECT ----> ISPFDEMO
GROUP ----> XXX      ----> A      ----> MASTER  ---->
TYPE ----> COBOL
MEMBER ----> TOP      (Blank or pattern for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ---->

LIST ID ----> LSTCOBOL_ (Blank for hardcopy listing)
SYSOUT CLASS ----> (If hardcopy requested)

COMPILER OPTIONS:
TEST ----> TEST (TEST or NOTEST)
OTHER ----> SOURCE,XREF,LINEC(73)

ADDITIONAL INPUT LIBRARIES:
----> 'ISPFTEST.FLAG.COBOL'
---->
  
```

Figure 138. Batch OS/VS COBOL Compile Panel

The Batch OS/VS COBOL Compile panel is shown in Figure 138. All the fields on this panel are explained in Chapter 2, "Libraries and Data Sets," except LIST ID, which is explained in "List Data Sets" on page 216, ADDITIONAL INPUT LIBRARIES, which is explained in "Input Data Sets" on page 215, and the following:

TEST If you plan to run COBOL interactive debug after you compile your program, enter TEST in the TEST field. Otherwise, enter NOTEST.

Note: Before you can specify TEST, you must have already used the Data Set utility (option 3.2) to allocate space for a symbolic debug data set. See "Symbolic Debug Data Sets" on page 242 if you need more information.

OTHER Enter any other options you need in the OTHER field. The available Batch OS/VS COBOL compiler options are:

ADV	DUMP	LCOL2	NAME	SPACE1	SYST
APOST	DUMP	LIB	NUM	SPACE2	TERM
BATCH	DYNAM	LINEC	OPTIMIZE	SPACE3	TRUNC
BUF	ENDJOB	LOAD	PMAP	STATE	VBREFM
CDECK	FDECK	LSTCOMP	PRINT	SUPMAP	VBSUM
CLIST	FLAGE	LSTONLY	RESIDENT	SXREF	VERB
COMPILE	FLAGW	LVL	SEQ	SYMDMP	XREF
COUNT	FLOW	L120	SIZE	SYNTAX	ZWB
DECK	LCOL1	L132	SOURCE		

Figure 139. Batch OS/VS COBOL Compiler Options

Chapter 11. LM Utilities (Option 8)

When you select the Library Management Utilities option (8) from the ISPF/PDF Primary Option Menu, the panel shown in Figure 175 is displayed. The options on this panel give you access to the Library Management Facility (LMF), which is an extension of the ISPF library concept.

Notes:

1. *To use the Library Management Utilities option, you must have preallocated the ISRCFIL control file.*
2. *For complete information on the use of the Library Management Utilities option, see Library Management.*

```

----- LIBRARY MANAGEMENT UTILITIES -----
OPTION ===> _

1  CONTROLS      - Specify library attributes:
                   Create, update, or delete promotion controls.
                   Predefine, update or delete member controls.
2  USER SET      - Create, update, or delete list of users authorized to
                   perform library functions.
3  DISTRIBUTION   - Create, update, or delete a distribution table to relate
                   different promotion hierarchies.
4  ACTIVATE      - Activate or deactivate promotion hierarchy controls.
5  REVIEW        - Browse or print activity and library controls
                   information.
6  ACTIVITY LOG   - Browse or remove entries from activity logs.
7  DS FULL ACTION - Specify actions to take when LMF data sets fill up.
T  TUTORIAL      - General information about the library management
                   utility facility.
  
```

Figure 175. Library Management Utilities Panel

LMF allows you to control the contents of your libraries and manage your development processes. Therefore, libraries that you control by using LMF are called *controlled libraries*.

LMF requires a library administrator to set up and maintain library controls. If you create the controls for a library for a new project, you automatically become the project administrator for that project.

Batch - VS FORTRAN Compiler (Option 5.3)

The available Batch VS FORTRAN compiler options are:

DECK	FREE	LINECOUNT	NAME	SOURCE	TEST
FIPS	GOSTMT	LIST	OBJECT	TERMINAL	XREF
FLAG	LANGVL	MAP	OPTIMIZE		

Figure 141. Batch VS FORTRAN Compiler Options

CHAPTER 12. IBM PRODUCTS (Option 9)

This chapter describes IBM Products, option 9 on the ISPF/PDF Primary Option Menu. This option lets you invoke other IBM program development products while using ISPF/PDF, but only if those products are installed and available on your system. The other IBM products supported are:

CSP/AD	Cross System Product/Application Development
CSP/AE	Cross System Product/Application Execution
INFO/SYS	Information/System
COBOL/SF-F	COBOL Structuring Facility foreground dialog
COBOL/SF-B	COBOL Structuring Facility background dialog

PL/I Optimizing Compiler (Option 5.5)

The Batch PL/I Optimizing Compile panel is shown in Figure 144.

```
----- BATCH PL/I OPTIMIZING COMPILE -----
COMMAND ---->

ISPF LIBRARY:
PROJECT ----> ISPFDEMO
GROUP ----> MYLIB      ----> MASTER      ----> FLAG      ---->
TYPE ----> PLI
MEMBER ----> TOPSEG      (Blank or pattern for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
DATA SET NAME ---->

LIST ID ----> _      (Blank for hardcopy listing)
SYSOUT CLASS ----> A      (If hardcopy requested)

COMPILER OPTIONS:
----> MACRO,XREF

ADDITIONAL INPUT LIBRARIES:
----> 'OURSYS.PLIMACS'
---->
```

Figure 144. Batch PL/I Optimizing Compile Panel

All the fields on this panel are explained in Chapter 2, “Libraries and Data Sets,” except LIST ID, which is explained in “List Data Sets” on page 216 , ADDITIONAL INPUT LIBRARIES, which is explained in “Input Data Sets” on page 215, and the following:

COMPILER OPTIONS

The available Batch PL/I optimizing compiler options are:

AGGREGATE	FLAG	LINECOUNT	NEST	SEQUENCE
ATTRIBUTES	FLOW	LIST	NOSPIE	SIZE
CHARSET	GONUMBER	LMESSAGE	NOSTAE	SOURCE
COMPILE	GOSTMT	MACRO	NUMBER	STMT
CONTROL	GRAPHIC	MAP	OBJECT	STORAGE
COUNT	IMPRECISE	MARGINI	OFFSET	SYNTAX
DECK	INCLUDE	MARGINS	OPTIMIZE	TERMINAL
DUMP	INSOURCE	MDECK	OPTIONS	XREF
ESD	INTERRUPT	NAME		

Figure 145. Batch PL/I Optimizing Compiler Options

Chapter 12. IBM Products (Option 9)

Option 9 provides an interface to IBM program development products other than ISPF/PDF. It displays the following panel (Figure 176), which lists other IBM products that are supported as ISPF dialogs:

```
----- ADDITIONAL IBM PROGRAM DEVELOPMENT PRODUCTS -----  
OPTION ----> _  
  
1 CSP/AD      - Cross System Product/Application Development  
2 CSP/AE      - Cross System Product/Application Execution  
3 INFO/SYS    - Information/System  
4 COBOL/SF-F  - COBOL Structuring Facility foreground dialog  
5 COBOL/SF-B  - COBOL Structuring Facility background dialog
```

Figure 176. Additional IBM Program Development Products Panel

When you select one of these products, one of the following occurs:

- If the product is installed and available on your system, PDF tries to invoke it. However, the only way PDF can determine whether a product is installed and available is to check for the existence of a single product-related panel in the panel library concatenation. No other check is made to ensure that the product is correctly installed or that it is completely available to you.
- If the product is not installed or available, PDF displays an informational panel that describes the product and shows how additional information can be obtained.

The following sections describe the options shown at the top of the Additional IBM Program Development Products panel.

Batch - Linkage Editor (Option 5.7)

Linkage Editor (Option 5.7)

The Batch Linkage Editor is invoked from the Batch Linkage Edit panel, shown in Figure 148.

```
----- BATCH LINKAGE EDIT -----
COMMAND ===>

ISPF LIBRARY:
  PROJECT ===> ISPFDEMO
  GROUP   ===> MYLIB   ===> MASTER   ===>
  TYPE    ===> OBJ
  MEMBER  ===> MAINPROG      (Blank or pattern for member selection list)

OTHER PARTITIONED DATA SET:
  DATA SET NAME ===>

LIST ID      ===> LLLL_      (Blank for hardcopy listing)
SYSOUT CLASS ===>           (If hardcopy requested)

LINKAGE EDITOR OPTIONS:
  TERM ===>                 (TERM or Blank)
  OTHER ===> MAP,LET,TEST,RENT,DCBS(3072)

ADDITIONAL INPUT LIBRARIES:      (LOAD libraries only)
  ===>
  ===>
```

Figure 148. Batch Linkage Edit Panel

All the fields on this panel are explained in Chapter 2, “Libraries and Data Sets,” except LIST ID, which is explained in “List Data Sets” on page 216 , ADDITIONAL INPUT LIBRARIES, which is explained in “Input Data Sets” on page 215, and the following:

TERM In the TERM field, enter TERM if you want PDF to generate a terminal data set. A *terminal data set* contains a synopsis of the error messages produced by the linkage editor. If the input data set is partitioned, the terminal data set name is:

prefix.userid.member.TERM

where “prefix” is your TSO prefix, if you have one and if it is different from your user ID, “userid” is your user ID, and “member” is the name of the member being assembled.

Note: Sequential data sets are invalid when using the Linkage Editor.

Leave the TERM field blank to avoid generating the terminal data set.

OTHER Enter any other options you need in the OTHER field.

CSP/AE (Option 9.2)

If the Cross System Product/Application Execution (CSP/AE) licensed program is not installed or available on your system, PDF displays the panel shown in Figure 178:

```
----- ADDITIONAL IBM PROGRAM DEVELOPMENT PRODUCTS -----  
COMMAND ----> _  
  
| Cross System Product/Application Execution |  
|               not installed               |  
-----  
  
CSP/AE is not currently available to your terminal session. The initial  
CSP/AE panel does not exist in your panel library. CSP/AE is not part of  
ISPF/PDF, but may be installed along with ISPF/PDF.  
  
CSP/AE is used to run end user applications developed using the CSP/AD  
program development environment. Using CSP/AE, an application developed on  
any supported system can execute on any other supported system.  
  
For more information on CSP you may contact IBM Marketing or order these  
publications:  
  
GH23-0500 : CSP/AD and CSP/AE General Information  
GV21-5113 : Cross System Product Set: Executive Overview  
SH23-0503 : CSP/AD and CSP/AE System Administration
```

Figure 178. Cross System Product/Application Execution Panel

VS COBOL II Interactive Debug (Option 5.10)

Before you can run VS COBOL II interactive debug in batch, you must first do the following, in the order shown:

1. Compile the program using the VS COBOL II compiler (option 4.2 or option 5.2) with the TEST and RESIDENT options.
2. Use the linkage editor (option 4.7 or option 5.7) to generate an output load module, which VS COBOL II interactive debug will use as input.

The VS COBOL II Interactive Debug panel is shown in Figure 150.

```
----- VS COBOL II INTERACTIVE DEBUG -----
COMMAND ===>

ISPF LIBRARY:
  PROJECT ===> ISPFDEMO
  GROUP   ===> XXX           (TYPE = LOAD assumed)
  MEMBER  ===> TOP           (Blank or pattern for member selection list)

OTHER PARTITIONED OR SEQUENTIAL DATA SET:
  DATA SET NAME ===>

LIST ID   ===> LSTCOBOL_     (Blank for hardcopy listing)
SYSOUT CLASS ===>           (If hardcopy requested)

DEBUG COMMAND DATA SET:
  ===>

ADDITIONAL INPUT LIBRARIES:
  ===>
  ===>
```

Figure 150. VS COBOL II Interactive Debug Panel

All the fields on this panel are explained in Chapter 2, "Libraries and Data Sets," except LIST ID, which is explained in "List Data Sets" on page 216 , ADDITIONAL INPUT LIBRARIES, which is explained in "Input Data Sets" on page 215, and the following:

Note: For VS COBOL II interactive debug, any additional input libraries that you enter to complete the search sequence must be LOAD libraries only.

DEBUG COMMAND DATA SET

In the DEBUG COMMAND DATA SET field, enter the name of the data set that contains the the DEBUG command that you want VS COBOL II interactive debug to execute during batch processing. See *VS COBOL II Release 2 Application Programming: Debugging Guide* for more information.

COBOL/SF-F (Option 9.4)

If the COBOL Structuring Facility (COBOL/SF) is not installed or available on your system, PDF displays the panel shown in Figure 180:

```
----- ADDITIONAL IBM PROGRAM DEVELOPMENT PRODUCTS -----
COMMAND ----> _

      COBOL Structuring Facility
      is not installed

COBOL Structuring Facility (COBOL/SF) is not installed on your system or is
not available on your terminal session. COBOL/SF is not part of
ISPF/PDF, but may be installed along with ISPF/PDF.

COBOL/SF (5668-786) transforms unstructured VS COBOL II programs into
structured VS COBOL II for improved understandability and maintainability.
It also produces an extensive Re-engineering Report of the structural
qualities of the input program and the output program.

For more information on COBOL/SF you may contact IBM Marketing or order
these publications:

GC34-4078 : COBOL Structuring Facility Licensed Program Specifications
SC34-4079 : COBOL Structuring Facility Re-engineering Concepts
SC34-4080 : COBOL Structuring Facility User's Guide and Reference
```

Figure 180. COBOL Structuring Facility Panel - Foreground Dialog

Batch - Member Parts List (Option 5.12)

CHAPTER 13. TUTORIAL (Option T)

This chapter describes Tutorial, option T on the ISPF/PDF Primary Option Menu. The Tutorial option provides you with online information about PDF.

Chapter 13. Tutorial (Option T)

The Tutorial option (T) provides you with immediate online reference and instruction on how to use PDF. You can invoke it by:

- Selecting the Tutorial option (T) from the ISPF/PDF Primary Option Menu
- Using the HELP command, described under “Using the HELP Command” on page 338.

Selecting the Tutorial Option (T)

To invoke the tutorial from the ISPF/PDF Primary Option Menu, type T on the COMMAND line and press the ENTER key. The first panel shown in Figure 182 on page 336 is displayed. This panel and the two that follow it explain how the tutorial is organized and how you can use it.

You can see the other two panels in the figure by pressing the ENTER key. The use of the ENTER key to scroll through the tutorial is explained in the next section.

Selecting a Tutorial Topic

The introductory pages described above are followed by a table of contents (Figure 183 on page 337) from which you can select a topic by:

- Entering the desired option letter or number
- Pressing the ENTER key to see the next topic.

Pressing the ENTER key is the same as entering the RIGHT scroll command. The tutorial scroll commands are explained in the introductory pages, shown in Figure 182 on page 336.

Note: The UP command displays the list of topics for the next level. A new topic can then be selected from the list. If you enter the UP command after viewing a portion of the tutorial sequentially and if no new topic is selected from the list, the tutorial resumes with the next sequential topic in the list when you press the ENTER key or enter the NEXT command.

Tutorial - Selecting the Tutorial Option (T)

----- ISPF TUTORIAL -----

COMMAND ----> _

ISPF PROGRAM DEVELOPMENT FACILITY

TUTORIAL

This tutorial provides on-line information about the features and operation of the ISPF program development facility (ISPF/PDF). You may view the tutorial sequentially, or you may choose selected topics from lists that are displayed on many of the tutorial pages.

The table of contents contains a list of major topics. Subsequent pages contain additional lists that lead you through more specific levels of detail. You can also select topics from the tutorial index.

The next two pages contain a description of how to use this tutorial.

Press ENTER key to proceed to the next page, or
Enter UP command to go directly to the table of contents, or
Enter END command to return to the primary option menu.

TUTORIAL ----- INTRODUCTION ----- TUTORIAL

COMMAND ----> _

You may view the tutorial sequentially by leaving the command/option field blank and repeatedly pressing the ENTER key. Alternatively, you may select topics from lists that are displayed on many of the tutorial pages. For example enter:

OPTION ----> 3 to select topic 3.

You may also enter one of the following commands on any tutorial page:

BACK or B - to back up to the previously viewed page.
SKIP or S - to skip the current topic and go on to the next topic.
UP or U - to display a higher level list of topics.
TOC or T - to display the table of contents.
INDEX or I - to display the tutorial index.

(continued on next page)

Figure 182 (Part 1 of 2). ISPF Tutorial and Introduction Panels

Tutorial - Selecting the Tutorial Option (T)

```
TUTORIAL ----- INTRODUCTION (CONTINUED) ----- TUTORIAL
COMMAND ==> _
```

You may use the following program function (PF) keys while viewing the tutorial:

HELP - to get help on how to use the tutorial.	HELP PF1 or PF13	END PF3 or PF15
END - to end the tutorial.		
UP - to display a higher level list of topics.		
DOWN - to go on to the next topic (skip).	UP PF7 or PF19	DOWN PF8 or PF20
LEFT - to display the previous page (back).		
RIGHT - to display the next page, which is the same as pressing ENTER.	LEFT PF10 or PF22	RIGHT PF11 or PF23

Figure 182 (Part 2 of 2). ISPF Tutorial and Introduction Panels

```
TUTORIAL ----- TABLE OF CONTENTS ----- TUTORIAL
OPTION ==> _
```

ISPF PROGRAM DEVELOPMENT FACILITY TUTORIAL

The following topics are presented in sequence, or may be selected by entering a one-character selection code in the option field:

- G GENERAL - General information about ISPF
- 0 ISPF PARMS - Specify terminal and user parameters
- 1 BROWSE - Display source data or output listings
- 2 EDIT - Create or change source data
- 3 UTILITIES - Perform utility functions
- 4 FOREGROUND - Invoke language processors in foreground
- 5 BATCH - Submit job for language processing
- 6 COMMAND - Enter TSO command or CLIST
- 7 DIALOG TEST - Perform dialog testing
- 8 LM UTILITY - Perform library administrator utility functions
- 9 IBM PRODUCTS- Use additional IBM program development products
- X EXIT - Terminate ISPF using log and list defaults

The following topics will be presented only if explicitly selected:

- A APPENDICES - Dynamic allocation errors and ISPF listing formats
- I INDEX - Alphabetical index of tutorial topics

Figure 183. Table of Contents Panel

Ending the Tutorial

To end the tutorial, enter either the END or RETURN command, which causes a return either to the ISPF/PDF Primary Option Menu or to the display on which you entered the HELP command.

Tutorial - Using the HELP Command

Using the HELP Command

If a short message appears in the upper-right corner of a panel, you can enter the HELP command to display a longer message that provides additional information. For more help, enter the HELP command again to display the appropriate section of the tutorial, based on what you were doing.

If you need information about using the tutorial, enter the HELP command again to display a one-page summary, shown in Figure 184. See the topics shown under “Selecting the Tutorial Option (T)” on page 335 for information about:

- Selecting a tutorial topic
- Ending the tutorial.

```
TUTORIAL ----- HELP TUTORIAL ----- TUTORIAL
COMMAND ----> _

in addition to topics which may be selected from individual tutorial
pages, you may enter one of the following in the command/option field
on any tutorial page:
  BACK or B - to back up to the previously viewed page.
  SKIP or S - to skip the current topic and go on to the next topic.
  UP or U - to display a higher level list of topics.
  TOC or T - to display the table of contents.
  INDEX or I - to display the tutorial index.

You may use the following keys whenever you are in the tutorial:
  ENTER - to display the next sequential page.
  HELP - to redisplay this page for "help" information.
  END - to terminate the tutorial.
  UP - to display a higher level list of topics (instead of typing UP ).
  DOWN - to skip to the next topic (instead of typing SKIP ).
  RIGHT - to display the next page (instead of pressing ENTER ).
  LEFT - to display the previous page (instead of typing BACK ).

Press ENTER to continue viewing the tutorial.
```

Figure 184. Help Tutorial Panel

Appendix A. Listing Formats

This appendix describes and displays the types of listings you can produce using PDF. The sample listings shown on the following pages are for illustration purposes only. They are not intended to be exact replicas, because printouts of PDF listings will vary according to the type of printer you are using.

Source and Index Listings

If autolist mode is on, the PDF editor automatically generates a source listing when you finish editing. You can also get source listings and index listings by using the following utility options:

Library (3.1)

P - Print member

Move/Copy (3.3)

CP - Copy and print

MP - Move and print

LP - Copy, lock, and print

PP - Promote and print

Data Set List (3.4)

P - Print data set list

PV - Print VTOC entries

Hardcopy (3.6)

PK - Print and keep data set

PD - Print and delete data set

Outlist (3.8)

P - Print job output

Source and Index Listings

Source Listings

Figure 185 shows a sample source listing.

○	PROJECT: ISPFDEMO	MEMBER: COINS	DATE: 86/07/30	○
○	GROUP: MYLIB	LEVEL: 01.04	TIME: 17:22	○
○	TYPE: PLI	USERID: HUNTER	PAGE: 01 OF 01	○
○	START COL	1-----2-----3-----4-----5-----6-----7-----8	MOD	○
○	COL		FLAGS	○
○	2	COINS:	00010001	○
○	4	PROCEDURE OPTIONS (MAIN);	00020000	○
○	6	DECLARE	00030000	○
○	8	COUNT FIXED BINARY (31) AUTOMATIC INIT (1),	00040000	○
○	8	HALVES FIXED BINARY (31),	00050000	○
○	8	QUARTERS FIXED BINARY (31),	00060003	○
○	8	DIMES FIXED BINARY (31),	00070000	○
○	8	NICKELS FIXED BINARY (31),	00080004	○
○	8	SYSPRINT FILE STREAM OUTPUT PRINT;	00090000	○
○	6	DO HALVES = 100 TO 0 BY -50;	00100000	○
○	8	DO QUARTERS = (100 - HALVES) TO 0 BY -25;	00110000	○
○	10	DO DIMES = ((100 - HALVES - QUARTERS)/10)*10 TO 0 BY -10;	00120000	○
○	12	NICKELS = 100 - HALVES - QUARTERS - DIMES;	00130000	○
○	12	PUT FILE(SYSPRINT) DATA(COUNT,HALVES,QUARTERS,DIMES,NICKELS);	00140000	○
○	12	COUNT = COUNT + 1;	00150000	○
○	10	END;	00160000	○
○	8	END;	00170000	○
○	6	END;	00180000	○
○	4	END COINS;	00190001	○
○				○

Figure 185. Sample Source Listing

Information at the top of the page includes project, group, type, and member name, current version and modification level, user ID, date and time that the listing was produced, and page number.

A column-positioning line is printed following the heading and preceding the actual data. The *start column* is printed to the left of each line, indicating the position of the first character in each line that is not a blank.

For ISPF library members with statistics, asterisks are either printed or not printed to the right of each line according to the setting of the modification flag, as follows:

- If the modification flag (columns 79-80) in the line is 00, no asterisks are printed.
- If the modification flag is nonzero but differs from the current modification level of the member, a single asterisk (*) is printed.
- If the modification flag is nonzero and has the same value as the current modification level of the member, two asterisks (**) are printed.

The asterisks allow you to scan the listing quickly for lines that were added or changed since the version was created (*) and for lines that were added or changed during the last update (**).

Index Listings

PDF provides index listings at your request through the X (Print index listing) option of the Library utility (3.1) or the Data Set List utility (3.4).

Index Listings for Source Libraries

Figure 186 shows a sample index listing for an ISPF library.

PROJECT: ISPFDEMO

GROUP: MYLIB

TYPE: PLI

DATE: 86/02/07

TIME: 17:22

PAGE: 001

GENERAL DATA:

VOLUME SERIAL: D00163

DEVICE TYPE: 3330

ORGANIZATION: PO

CREATION DATE: 85/12/08

EXPIRATION DATE: **NONE**

GENERAL DATA:

RECORD FORMAT: FB

RECORD LENGTH: 80

BLOCK SIZE: 1,600

1ST EXTENT SIZE: 20

SECONDARY QUAN: 0

CURRENT ALLOCATION:

20 TRACKS

1 EXTENT

10 DIRECTORY BLOCKS

CURRENT UTILIZATION:

5 TRACKS

1 EXTENT

2 DIRECTORY BLOCKS

10 MEMBERS

MEMBER NAME	VERS.MOD LEVEL	CREATION DATE	DATE AND TIME LAST MODIFIED	CURRENT NO. LINES	INITIAL NO. LINES	MODIFIED NO. LINES	USER ID
ACCOUNT	01.00	86/01/10	86/01/10 17:07	21	21	0	HUNTER
ACCT1	01.01	86/01/09	86/01/23 14:52	99	193	0	BECKETT
ACCT2	01.00	86/01/09	86/01/10 17:07	20	20	0	BECKETT
COINS	01.04	86/01/24	86/01/25 16:20	19	19	4	JOSLIN
COMPX	01.00	86/01/09	86/01/10 17:08	44	44	0	JOSLIN
COMPY	01.01	86/01/14	86/01/15 12:30	13	13	1	FISHER
DCLS	01.00	86/01/23	86/01/24 15:14	20	20	0	SAMLEE
LISTOUT	01.02	86/01/23	86/01/24 15:00	17	13	6	SAMLEE
MAIN	01.00	86/01/09	86/01/10 17:08	4	4	0	KIRK
TESTDIR	01.02	86/01/23	86/02/06 17:04	30	43	10	LAIDLAW
MAXIMUMS:	01.04	86/04/24	86/05/06 17:04	99	193	10	
TOTALS:				287	390	21	

END OF MEMBER LIST

Figure 186. Sample Index Listing - Source Library

The heading information includes:

- Project, library, and type
- Date and time the listing was produced
- Page number.

This is followed by general information about the data set, including current space allocation and utilization. Following this, the member name and statistics are printed for each member in the data set, arranged in alphabetic order. For sequential data sets, the index listing contains just the general information.

Source and Index Listings

Index Listings for Load Libraries

An index listing for an OBJ library is similar to an index listing for a source library, except that no statistics are maintained. A sample index listing for a LOAD library is shown in Figure 187. Here, the module attributes are printed to the right of each member name.

PROJECT: ISPFDEMO		DATE: 86/02/07						
GROUP: MYLIB		TIME: 17:22						
TYPE: LOAD		PAGE: 001						
GENERAL DATA:		GENERAL DATA:	CURRENT ALLOCATION:	CURRENT UTILIZATION:				
VOLUME SERIAL:	D00172	RECORD FORMAT:	U	88 BLOCKS				
DEVICE TYPE:	3330	RECORD LENGTH:	0	12 EXTENTS				
ORGANIZATION:	PO	BLOCK SIZE:	3,072	4 DIRECTORY BLOCKS				
CREATION DATE:	85/12/08	1ST EXTENT SIZE:	44	8 MEMBERS				
EXPIRATION DATE:	**NONE**	SECONDARY QUAN:	4					
MEMBER NAME	ALIAS OF	SSI	LOAD MODULE SIZE (HEX) (DEC)	ENTRY POINT (HEX)	TTR (HEX)	AUTH CODE	MODULE	----- ATTRIBUTES -----
ISPFEDIT			000148 328	000000	000006	00	FO	RF RN RU
ISPFMAIN			004F98 20,376	000000	000000	00	FO	RF RN RU
ISPFMD			0016E8 5,864	000000	000204	00	FO	RF RN RU
ISPFSUBS			024698 149,144	000000	000211	00	FO	RF RN RU
ISPFIBLS			002470 9,328	000000	00110B	00	FO	
ISPFUDA			0046D8 18,136	000000	00120A	00	FO	RF RN RU
ISPFURS			000FCD 4,032	000000	00140A	00	FO	RF RN RU
ISPF3277			000548 1,352	000000	001504	00	FO	RF RN RU
TOTALS:			00032EB0 208,560					
END OF MEMBER LIST								

MODULE ATTRIBUTE CODES:								
FO	PROCESS ONLY BY F LEVEL LINKAGE EDITOR							
NX	NOT EXECUTABLE							
OL	ONLY LOADABLE							
OV	IN OVERLAY STRUCTURE							
RF	REFRESHABLE							
RN	REENTERABLE							
RU	REUSABLE							
SC	SCATTER FORMAT							
TS	MODULE TO BE TESTED							

Figure 187. Sample Index Listing - Load Library

ISPF Log Listings

Figure 188 shows a sample ISPF log listing. The log contains a message for each significant user action, such as saving edited data, moving members from one data set to another, or submitting a batch job.

TIME	*** ISPF TRANSACTION LOG ***	USERID: WALTERS	DATE: 86/06/29	PAGE: 1
17:11	START OF ISPF SESSION LOG #64 -----			
17:12	UTILITY - ALLOCATE - ISPFDEMO.NEWLIB.PLI ALLOCATED ON VOLUME WORK97			
17:13	UTILITY - MOVE/COPY- ISPFDEMO.MYLIB.PLI(ACCOUNT) COPIED			
17:13	TO ISPFDEMO.NEWLIB.PLI(ACCTXX)			
17:13	UTILITY - MOVE/COPY- ISPFDEMO.MYLIB.PLI(ACCT1) COPIED			
17:13	TO ISPFDEMO.NEWLIB.PLI(ACCTYY)			
17:13	UTILITY - MOVE/COPY- ISPFDEMO.MYLIB.PLI(ACCT2) COPIED			
17:13	TO ISPFDEMO.NEWLIB.PLI(ACCTZZ)			
17:13	UTILITY - MOVE/COPY- ISPFDEMO.MYLIB.PLI(COINS) COPIED			
17:13	TO ISPFDEMO.NEWLIB.PLI(COINS)			
17:14	EDIT - SAVE - ISPFDEMO.NEWLIB.PLI(COINS) - MEMBER SAVED			
17:15	EDIT - CREATE - ISPFDEMO.NEWLIB.PLI(ACCTNEW) - MEMBER CREATED			
17:15	EDIT - SAVE - ISPFDEMO.NEWLIB.PLI(ACCTXX) - MEMBER SAVED			
17:15	EDIT - SAVE - ISPFDEMO.NEWLIB.PLI(ACCTYY) - MEMBER SAVED			
17:16	EDIT - SAVE - ISPFDEMO.NEWLIB.PLI(ACCTZZ) - MEMBER SAVED			
17:17	UTILITY - DELETE - ISPFDEMO.NEWLIB.PLI(ACCTZZ) DELETED			
17:17	TSO - COMMAND - LISTC			
17:17	UTILITY - ALLOCATE - ISPFDEMO.NEWLIB.OBJ ALLOCATED ON VOLUME WORK98			
17:19	JOB - WALT8 - PLI OPT COMPILE -- TO ISPFDEMO.NEWLIB.OBJ(COINS)			
17:19	JOB - WALT8 - SUBMITTED			
17:20	END OF ISPF SESSION LOG #64 -----			

Figure 188. Sample ISPF Log Listing

Member List Listings

This section shows samples of member list listings created by the SAVE command. With this command, you can create listings for both source and load libraries.

The following sample listings show the format used when you do not specify a list ID.

Member List Listings for Source Libraries

Figure 189 shows a sample member list listing for a source library. These listings contain the relative block address of each member, shown in hexadecimal format, and other characteristics when available.

DATASET: FBROWN.TEST1.SCRIPT						DATE: 87/04/14			
						TIME: 23:51			
						PAGE: 001			
MEMBER	TTR	VERS.	MOD	CREATION	DATE AND TIME	CURRENT	INITIAL	MODIFIED	USER
NAME	(HEX)	LEVEL		DATE	LAST MODIFIED	NO. LINES	NO. LINES	NO. LINES	ID
ABEOLD	00001A	01.00		87/03/08	87/03/08 18:48	46	46	0	FBROWN
DATA	000018	01.00		87/02/24	87/02/24 17:18	5	5	0	FBROWN
WORDSOLD	000016	01.00		87/02/14	87/02/14 10:10	8	8	0	FBROWN
END OF MEMBER LIST									

Figure 189. Sample Member List Listing for a Source Library

Member List Listings for Load Libraries

Figure 190 shows a sample member list listing for a source library. These listings contain the size of each load module, shown in hexadecimal format, and other characteristics when available.

DATASET: FBROWN.TEST1.LOAD										DATE: 87/04/21
										TIME: 16:37
										PAGE: 001
MEMBER NAME	ALIAS OF	SSI	LOAD MODULE (HEX)	SIZE (DEC)	ENTRY POINT (HEX)	TTR (HEX)	AUTH CODE	MODULE ATTRIBUTES		
#ASTBANK		40820565	002160	8,544	000000	00011F	00	FO	RN	RU
CALENDAR		73500137	000380	896	000000	000127	00	FO		
CLR	CLRGEX	50250830	00B3E8	46,056	0011FC	000922	00	FO	RN	RU
CLRALLOC		90521058	0005C8	1,480	000000	000203	00	FO	RN	RU
CLRBLIB		72280809	000498	1,176	000000	00020A	00	FO		
CLRB0BJ		00110213	004820	18,464	002840	000213	00	FO		
CLRBSYM1		00110221	0060A0	24,736	0034E0	000210	00	FO		
CLRCDFDM			001920	6,432	000000	000308	00	FO	RU	TS
CLRCDFE			000028	40	000000	000312	00	FO	RU	TS
CLRCID01		93120466	003198	12,696	000000	00031A	00	FO	RN	RU
CLRCIPHR		90760481	000388	904	000000	000406	00	FO		
CLRFICHE		00110206	004958	18,776	002958	000908	00	FO		
CLRFLIP		02870484	0004D8	1,240	000000	000912	00	FO	RN	RU
CLRCENER		40400008	000A10	2,576	000000	00091A	00	FO		
CLRGEX		50250830	00B3E8	46,056	0011FC	000922	00	FO	RN	RU
CLRIDSMV		62290360	000E68	3,688	000000	000B04	00	FO	RN	RU
CLRIDSX		40400009	001CE8	7,400	000000	000B0F	00	FO	RN	RU
CLRIDEX		41940002	002128	8,488	000000	000B17	00	FO		
CLRIDERR		62250392	0000E0	224	000000	000B1F	00	FO		
CLRIPX			003030	12,336	000000	000C02	00	FO		
CLRJBLOG		93510170	0011B0	4,528	000000	000C0C	01	FO	RN	RU
CLRLGOPY		40430031	001790	6,032	000100	000C14	00	FO		
CLRLGSTX		02670226	001208	4,616	000000	000C20	00	FO	RN	RU
CLRLGUTL		32560116	000E88	3,720	000000	000D04	00	FO	RN	RU
CLRLPINT		32442006	001498	5,272	000000	000D0C	00	FO		
CLRLSDDS		81770527	000E80	3,712	000000	000D13	00	FO	RN	RU
END OF MEMBER LIST										

Figure 190. Sample Member List Listing for a Load Library

Member List Listings

Formats for Member List Listings

The following is the format used by the SAVE command to create a member list listing for a source library. The members of a source library have formatted records (RECFM ≠ U).

Starting Column	Length in Characters	Description
4	8	Member name
19	6	Relative block address in hexadecimal format
25	2	Version number
28	2	Modification level
31	8	Creation date
40	8	Date last modified
49	5	Time last modified
55	5	Current number of lines
61	5	Initial number of lines
67	5	Number of modified lines
73	7	User ID

Figure 191. Format of Source Library Member List Listing

The following is the format used by the SAVE command to create a member list listing for a load library. The members of a load library have unformatted records (RECFM = U).

Starting Column	Length in Characters	Description
4	8	Member name
24	6	Load module size in hexadecimal format
31	6	Load module relative block address in hexadecimal format
38	8	Alias
47	2	Authorization code
50	26	Load module attributes

Figure 192. Format of Source Library Member List Listing

Data Set List Listings

The sample listing in Figure 193 shows the format used when you do not specify a data set list ID.

LISTING OF DATA SETS BEGINNING WITH FBROWN										DATE: 87/04/14
										TIME: 23:06
										PAGE: 1
DATA SET NAME	VOLUME	ORG	RECFM	LRECL	BLKSZ	TRKS	%USED	XT	CREATED	
FBROWN.CLIST	HSMPOC	PO	VB	255	6160	1	100	1	1985/10/03	
FBROWN.DB2X.DXT210.DVRIMEXE	HSMPOF	PO	FB	80	3120	15	6	1	1987/03/05	
FBROWN.DB2X.DXT210.DVRJEDIE	HSMPOF	PO	FB	80	6160	15	13	1	1987/03/05	
FBROWN.EXAMPLE.DATASETS	HSMPO7	PS	FB	132	6072	19	5	1	1987/04/14	
FBROWN.FWB.CLIST	MIGRAT									
FBROWN.FWB.SCRIPT	HSMPO7	PO	VB	80	3120	1	100	1	1986/03/19	
FBROWN.FWB.TABLES	MIGRAT									
FBROWN.ISPTABL	HSMPOF	PO	FB	80	3120	1	100	1	1987/02/02	
FBROWN.LOG.MISC	MIGRAT									
FBROWN.MASTER.ISPPROF	MIGRAT									
FBROWN.PRIVATE.CLIST	HSMPOC	PO	VB	255	6160	1	100	1	1986/10/03	
FBROWN.PRIVATE.LOAD	HSMPOC	PO	U	0	6144	1	100	1	1986/10/03	
FBROWN.PRIVATE.MSGS	HSMPOC	PO	FB	80	6160	1	100	1	1986/10/03	
FBROWN.PRIVATE.PANELS	HSMPOC	PO	FB	80	6160	2	100	1	1986/10/03	
FBROWN.PRIVATE.SKELS	HSMPOC	PO	FB	80	6160	1	100	1	1986/10/03	
FBROWN.PRIVATE.TABLES	HSMPOC	PO	FB	80	6160	2	100	1	1986/10/03	
FBROWN.SMALL.FIXED	HSMPOB	PO	FB	60	600	1	100	1	1987/04/09	
FBROWN.TEST.CLIST	MIGRAT									
FBROWN.TEST.MSGS	MIGRAT									
FBROWN.TEST.SCRIPT	MIGRAT									
FBROWN.TEST1.SCRIPT	MIGRAT									
FBROWN.TEST2.SCRIPT	MIGRAT									

Figure 193. Sample Data Set List Listing

The following is the format of the data set list written by the SAVE command when a data set list ID is specified:

Starting Column	Length in Characters	Description
1	44	Data set name
46	6	Volume
53	4	Data set organization
58	5	Data set record format
64	5	Data set logical record length
70	5	Data set block size
76	6	Data set size in tracks
83	3	Percentage of used tracks
87	3	Number of extents used
91	8	Device type
100	10	Creation date
111	10	Expiration date
122	10	Last Reference date

Figure 194. Format of Data Set List Listing

SuperC Listings

The SuperC program lets you produce four types of listings, differing generally depending on the compare type, process options, and process statements used. They range from OVSUM, which shows none of the source data, to LONG, which shows all of it.

Note: Listing types are not applicable for the FILE compare type. For information about FILE compare listings, see “Effect of FILE Compare Type on SuperC Listings” on page 358.

OVSUM Lists only an overall summary of the results of the comparison without showing the differences themselves.

For example, an OVSUM listing of a LINE comparison shows how many lines matched, how many were reformatted, how many were inserted, and so forth.

DELTA Lists only the differences between the source data sets or members being compared, followed by overall summary results. Differences are flagged to the left of each output line.

For example, a DELTA listing of a LINE comparison shows only the individual lines in each data set or member that are different.

CHNG Same as the DELTA listing, plus up to 10 matching lines, words, or bytes before and after the differences. This shows the differences within the context of the surrounding lines.

For example, suppose a comparison of two data sets shows that each has 15 lines that match, followed by one or more lines that are different, followed by 15 more lines that match. A CHNG listing of a LINE comparison of these data sets would show the lines in each data set or member that are different, plus 10 of the 15 lines that match, both before and after.

LONG Same as the CHNG listing, except this listing shows the *entire* “new” data set, plus any data from the “old” data set that is not in the “new.”

If we borrow the example used in the CHNG listing definition, a LONG listing would include all 15 matching lines, not just 10.

The following sections begin by explaining the information displayed in the titles lines and at the bottom of the listings. Then, each listing type is discussed in more detail, with examples of listings and descriptions of the listing sections.

The examples were created by comparing two PDSs. The first, FBROWN.TEST1.SCRIPT, is considered the “old,” or original, data set and contains members DATA, ABEOLD, and WORDSOLD.

FBROWN.TEST2.SCRIPT, the “new,” or changed, data set, contains members DATA, ABENEW, and WORDSNEW.

DATA, as the only member name common to both data sets, is the only member compared. Figure 195 shows the source data for both the “old” and “new” versions of DATA.

```

EDIT ---- FBROWN.TEST1.SCRIPT(DATA) - 01.02 ----- COLUMNS 001 072
COMMAND ==> SCROLL ==> PAGE
***** TOP OF DATA *****
000100 This line is reformatted; the spacing in the "new" member differs.
000200 This line is the same in both members.
000300 This line differs from the text in the "new" member.
000400 This line is the same in both members.
***** BOTTOM OF DATA *****

EDIT ---- FBROWN.TEST2.SCRIPT(DATA) - 01.03 ----- COLUMNS 001 072
COMMAND ==> SCROLL ==> PAGE
***** TOP OF DATA *****
000100 This line is reformatted; the spacing in the "new" member differs.
000200 This line is the same in both members.
000300 This line is different from the text in the "old" member.
000400 This line is the same in both members.
000500 This line is in the "new" member, but not in the "old".
***** BOTTOM OF DATA *****

```

Figure 195. Split Screen Showing Source Data for SuperC Listings

SuperC Listing Title Lines

SuperC listings contain two title lines. Figure 196 on page 350 shows an example. The first title line shows the:

- Name of the SuperC program, which is described as a file/line/word/byte compare program
- Version number of the program and the date this version went into effect.
- Date and time the listing was produced. The date is shown according to the format used by your country. The time is shown in 24-hour format, with the hours and minutes separated by a period.
- Page number.

The second title line shows you the names of the “new” and “old” data sets that were compared when the listing was created.

SUPERF - FILE/LINE/WORD/BYTE COMPARE PGM - V2.7(87/03/27)		87/04/15	20.24	PAGE	1
NEW:	FBROWN.TEST2.SCRIPT(DATA)	OLD:	FBROWN.TEST1.SCRIPT(DATA)		
LISTING OUTPUT SECTION (LINE COMPARE)					
ID	SOURCE LINES		TYPE	LEN	N-LN# O-LN#
1	-----1-----2-----3-----4-----5-----6-----7-----8				
RN-	This line is reformatted; the spacing in the "new" member differs.	00010000	RFM-	1	00001 00001
RO-	This line is reformatted; the spacing in the "new" member differs.	00010000			
			MAT-	1	
I-	This line is different from the text in the "old" member.	00030000	RPL-	1	00003 00003
O-	This line differs from the text in the "new" member.	00030000			
			MAT-	1	
I-	This line is in the "new" member, but not in the "old".	00050000	INS-	1	00005 00004
SUPERF - FILE/LINE/WORD/BYTE COMPARE PGM - V2.7(87/03/27)		87/04/15	20.24	PAGE	2
NEW:	FBROWN.TEST2.SCRIPT	OLD:	FBROWN.TEST1.SCRIPT		
MEMBER SUMMARY LISTING (LINE COMPARE)					
CHNG	MEMBERS-COMPARED	NEW-PROC	OLD-PROC	MATCHED	NEW-INS OLD-DEL REFM-LN N-DP-LN O-DP-LN
**	DATA	5	4	2	2 1 1
1	TOTAL MEMBER(S) PROCESSED AS A PDS				
1	TOTAL MEMBER(S) PROCESSED HAD CHANGES				
0	TOTAL MEMBER(S) PROCESSED HAD NO CHANGES				
2	TOTAL NEW FILE MEMBER(S) NOT PAIRED				
2	TOTAL OLD FILE MEMBER(S) NOT PAIRED				
SUPERF - FILE/LINE/WORD/BYTE COMPARE PGM - V2.7(87/03/27)		87/04/15	20.24	PAGE	3
NEW:	FBROWN.TEST2.SCRIPT	OLD:	FBROWN.TEST1.SCRIPT		
MEMBER SUMMARY LISTING (LINE COMPARE)					
NON-PAIRED NEW FILE MEMBERS		1	NON-PAIRED OLD FILE MEMBERS		
ABENEF		1	ABEOLD		
WORDNEW		1	WORDOLD		
SUPERF - FILE/LINE/WORD/BYTE COMPARE PGM - V2.7(87/03/27)		87/04/15	20.24	PAGE	4
NEW:	FBROWN.TEST2.SCRIPT	OLD:	FBROWN.TEST1.SCRIPT		
PDS LINE OVERALL TOTALS					
2	NUMBER OF LINE MATCHES	3	TOTAL CHANGES (PAIRED+NONPAIRED CHNG)		
1	REFORMATTED LINES	2	PAIRED CHANGES (REFM+PAIRED INS/DEL)		
2	NEW FILE LINE INSERTIONS	1	NON-PAIRED INSERTS		
1	OLD FILE LINE DELETIONS	0	NON-PAIRED DELETES		
5	NEW FILE LINES PROCESSED				
4	OLD FILE LINES PROCESSED				
LISTING-TYPE = DELTA		COMPARE-COLUMNS =	1:72	LONGEST-LINE = 80	
PROCESS OPTIONS USED: SEQ(DEFAULT)					

NW: 15457 DocId: 70002077 Page 744

SUPERC - FILE/LINE/WORD/BYTE COMPARE PGM - V2.7(87/03/27) 87/04/30 13.46 NEW: FBROWN.TEST2.SCRIPT(DATA) OLD: FBROWN.TEST1.SCRIPT(DATA)									
PAGE 1									
LISTING OUTPUT SECTION (WORD COMPARE)									
1D SOURCE LINES (COMPARED COLUMNS)									
								N-LN#	O-LN#
This line is reformatted; the spacing in the "new" member differs.								00001	00001
This line is the same in both members.								00002	00002
MC-This line is the same from the text in the "old" member.								00003	00003
IC- is different "old"								00003	00003
DC- differs "new"								00003	00003
This line is the same in both members.								00004	00004
I-This line is in the "new" member, but not in the "old".								00005	00004
SUPERC - FILE/LINE/WORD/BYTE COMPARE PGM - V2.7(87/03/27) 87/04/30 13.46 NEW: FBROWN.TEST2.SCRIPT OLD: FBROWN.TEST1.SCRIPT									
PAGE 2									
MEMBER SUMMARY LISTING (WORD COMPARE)									
CHNG MEMBERS-COMPARED	NEW-PROC	OLD-PROC	MATCHED	NEW-INS	OLD-DEL	N-DP-LN	O-DP-LN	N-LN-PRC	LN-INS
50	37	35	15	2		5	2	4	1
1 TOTAL MEMBER(S) PROCESSED AS A PDS 1 TOTAL MEMBER(S) PROCESSED HAD CHANGES 0 TOTAL MEMBER(S) PROCESSED HAD NO CHANGES 2 TOTAL NEW FILE MEMBER(S) NOT PAIRED 2 TOTAL OLD FILE MEMBER(S) NOT PAIRED									

SUPERC - FILE/LINE/WORD/BYTE COMPARE PGM - V2.7(87/03/27)		87/04/30	13.46	PAGE	3
NEW:	FBROWN.TEST2.SCRIPT	OLD:	FBROWN.TEST1.SCRIPT		
MEMBER SUMMARY LISTING (WORD COMPARE)					
NON-PAIRED NEW FILE MEMBERS		NON-PAIRED OLD FILE MEMBERS			
ABENew		ABEOld			
WORDSNEW		WORDSOld			
SUPERC - FILE/LINE/WORD/BYTE COMPARE PGM - V2.7(87/03/27)		87/04/30	13.46	PAGE	4
NEW:	FBROWN.TEST2.SCRIPT	OLD:	FBROWN.TEST1.SCRIPT		
PDS WORD OVERALL TOTALS					
35 NUMBER OF WORD MATCHES		15 TOTAL CHANGES (PAIRED+NONPAIRED CHNG)			
15 NEW FILE WORD INSERTIONS		2 NEW FILE LINES CHANGED/INSERTED			
2 OLD FILE WORD DELETIONS		1 OLD FILE LINES CHANGED/DELETED			
50 NEW FILE WORDS PROCESSED		5 NEW FILE LINES PROCESSED			
37 OLD FILE WORDS PROCESSED		4 OLD FILE LINES PROCESSED			
LISTING-TYPE = LONG		COMPARE-COLUMNS = 1:80		LONGEST-LINE = 80	
PROCESS OPTIONS USED: SEQ					

Figure 197. LONG Listing of a WORD Comparison of Two PDSs

Listing Output Section

The Listing Output Section shows any lines, words, or bytes that were found to be different, depending on the compare type being used. For a WORD comparison, the source lines that appear in the listing show only the columns of data that were really compared. See Figure 197 on page 351 for an example and “CMPCOLM - Compare Columns” on page 396 for more information.

The Listing Output Section flags changes by using:

- Listing prefix codes
- Type of difference codes.

Listing Prefix Codes

SuperC output lines are flagged with the following prefix codes listed under the ID column:

blank	<i>Match</i> —No prefix code means the data is the same in both data sets.
I	<i>Insert</i> —Data that is in the “new” data set, but not in the “old” data set.
D	<i>Delete</i> —Data that is in the “old” data set, but not in the “new” data set.
DR	<i>Delete Replace</i> —For BYTE compare listings only. A byte(s) in the “old” data set that was replaced by the byte(s) shown in the preceding insert (I) line.
MC	<i>Match Compose</i> —For WORD compare listings only. A line containing words that match. The line may also contain spaces to show the relationship between the matching words and any inserted or deleted words. Inserted and deleted words are shown in following insert compose (IC) and delete compose (DC) lines.
IC	<i>Insert Compose</i> —For WORD compare listings only. A line containing words from the “new” data set that are not in the “old” data set. This line usually follows a match compose (MC) line.
DC	<i>Delete Compose</i> —For WORD compare listings only. A line containing words from the “old” data set that are not in the “new” data set. This line usually follows a match compose (MC) or insert compose (IC) line.
IM	<i>Insert Matching</i> —For compare listings created by using the FMVLNS (flag moved lines) process option. A line in the “new” data set that also appears in the “old” data set, but has been moved. If the line was reformatted, this is indicated by a flag to the right of the listing.

- DM** *Delete Matching*—For compare listings created by using the FMVLNS (flag moved lines) process option. A line in the “old” data set that also appears in the “new” data set, but has been moved. If the line was reformatted, this is indicated by a flag to the right of the listing.
- RN** *Reformat New*—For LINE compare listings only. A reformatted line in the “new” data set. This line contains the same data as the “old” data set line, but with different spacing.
- RO** *Reformat Old*—For LINE compare listings only. A line in the “old” data set that has been reformatted in the “new” data set. To keep this line out of your output listing, use the DLREFM (delete reformatted line) process option.
- |** *Change Bar*—For compare listings created by using the GWCBL (generate WORD change bar listing) process option. A change bar showing that words were either inserted or deleted. Figure 198 on page 354 provides an example.

Type of Difference Codes

At the far right of some listings are headings that provide additional information about the types of differences SuperC has found. Headings you may see are:

- TYPE** An abbreviation that identifies the type of difference:
- MAT =** Matched lines.
- RFM =** Reformatted lines.
- RPL =** Replaced lines.
- INS =** Lines that are in the “new” data set, but not in the “old” data set.
- DEL =** Lines that are in the “old” data set, but not in the “new” data set.
- IMR =** Lines in the “new” data set that have been moved from where they were in the “old” data set and reformatted. Listing should show a matching DMR = flag for a line in the “old” data set.
- DMR =** Lines in the “old” data set that have been moved and reformatted in the “new” data set. Listing should show a matching IMR = flag for a line in the “new” data set.
- IMV =** Lines in the “new” data set that have been moved from where they were in the “old” data set. Listing should show a matching DMV = flag for a line in the “old” data set.

SuperC Listings

SUPERC - FILE/LINE/WORD/BYTE COMPARE PGM - V2.7(03/17/87)		87/04/24	15.50	PAGE	1
NEW:	FBROWN.TEST2.SCRIPT(DATA)	OLD:	FBROWN.TEST1.SCRIPT(DATA)		
LISTING OUTPUT SECTION (WORD COMPARE)					
ID	SOURCE LINES				
1	-This line is reformatted; the spacing in the "new" member differs.				
1	-This line is the same in both members.				
1	-This line is different from the text in the "old" member.				
1	-This line is the same in both members.				
1	-This line is in the "new" member, but not in the "old".				
SUPERC - FILE/LINE/WORD/BYTE COMPARE PGM - V2.7(03/17/87)		87/04/24	15.50	PAGE	2
NEW:	FBROWN.TEST2.SCRIPT	OLD:	FBROWN.TEST1.SCRIPT		
MEMBER SUMMARY LISTING (WORD COMPARE)					
CHNG	MEMBERS-COMPARED	NEW-PROC	OLD-PROC	MATCHED	NEW-INS
OLD-DEL	N-DP-LN	O-DP-LN	N-LN-PRC	LN-INS	O-LN-PRC
LN-DEL					
** DATA	55	40	37	17	2
1	TOTAL MEMBER(S) PROCESSED AS A PDS				
1	TOTAL MEMBER(S) PROCESSED HAD CHANGES				
0	TOTAL MEMBER(S) PROCESSED HAD NO CHANGES				
2	TOTAL NEW FILE MEMBER(S) NOT PAIRED				
2	TOTAL OLD FILE MEMBER(S) NOT PAIRED				

SUPERC - FILE/LINE/WORD/BYTE COMPARE PGM - V2.7(03/17/87)		87/04/24	15.50	PAGE	3
NEW:	FBROWN.TEST2.SCRIPT	OLD:	FBROWN.TEST1.SCRIPT		
MEMBER SUMMARY LISTING (WORD COMPARE)					
NON-PAIRED NEW FILE MEMBERS			NON-PAIRED OLD FILE MEMBERS		
ABENEW			ABEOLD		
WORDSNEW			WORDOLD		
SUPERC - FILE/LINE/WORD/BYTE COMPARE PGM - V2.7(03/17/87)		87/04/24	15.50	PAGE	4
NEW:	FBROWN.TEST2.SCRIPT	OLD:	FBROWN.TEST1.SCRIPT		
PDS WORD OVERALL TOTALS					
37	NUMBER OF WORD MATCHES	17	TOTAL CHANGES (PAIRED+NONPAIRED CHNG)		
17	NEW FILE WORD INSERTIONS	3	NEW FILE LINES CHANGED/INSERTED		
2	OLD FILE WORD DELETIONS	2	OLD FILE LINES CHANGED/DELETED		
55	NEW FILE WORDS PROCESSED	5	NEW FILE LINES PROCESSED		
40	OLD FILE WORDS PROCESSED	4	OLD FILE LINES PROCESSED		
1	FALSE MATCH(S) CORRECTED				
LISTING-TYPE = LONG COMPARE-COLUMNS = 1:80 LONGEST-LINE = 80					
PROCESS OPTIONS USED: GWCBL					

Figure 198. WORD Comparison of Two PDSs Using Change Bar to Show Differences

DMV = Lines in the “old” data set that have been moved in the “new” data set. Listing should show a matching IMV = flag for a line in the “new” data set.

LEN The number of consecutive lines flagged by one of the TYPE abbreviations.

N-LN# and O-LN#

The line number(s) where the flagged line appears in either or both data sets. For insertions, the O-LN# column shows where the line would have appeared in the “old” data set. For deletions, the N-LN# column shows where the line would have appeared in the “new” data set.

Member Summary Listing

The information shown in the Member Summary Listing depends on the compare type used. For example, PAGE 2 in Figure 198 on page 354 shows the number of words that were processed, matched, inserted, deleted, reformatted, and given “don’t process” status by a process option or statement, such as the DPLINE process option.

This part also shows how many members were processed, had changes, had no changes, and were not paired, with the last category provided for both “new” and “old” members.

As the sample listing shows, one member was processed and had changes. It also shows that both PDSs contain two *non-paired* members. The non-paired heading means that these members were not included in the comparison because they were not paired with members in the other PDS. The names of these members are shown in the second part of the Member Summary Listing, PAGE 3 of Figure 198 on page 354.

Summary and Statistics/PDS Overall Totals

Many SuperC listings contain an overall summary showing totals that are compiled for the compare type being used. For comparisons of sequential data sets or membered PDSs, this summary is called:

XXXX COMPARE SUMMARY AND STATISTICS

However, for comparisons of complete PDSs, as PAGE 4 of Figure 198 on page 354 shows, the summary is called:

PDS XXXX OVERALL TOTALS

In both headings, the “XXXX” is replaced by the compare type being used, such as WORD. As the sample listing shows, the left side of a LONG listing of a WORD comparison shows how many words were matched, inserted, deleted, and processed. Several of these categories are related. For instance, the number of lines processed in the “new” data set should be equal to the number of line matches, the reformatted lines, and the new file line insertions. The number of processed lines does not include lines given “don’t process” status by a process option or statement, such as the DPLINE process option.

The left side of the sample listing also shows that the SuperC program corrected a “false match,” which it occasionally may detect on the first pass and reclassify as an insert/delete pair. The redesignation is correct, but may have caused the process to overlook an alternate match. However, this condition should not occur frequently, which means your comparison results probably will not be affected. The SuperC program never incorrectly calls a data unit (file, line, word, or byte) a match when it should be a mismatch.

On the right side, the listing shows the total number of changes, the number of lines classified as changed/inserted and changed/deleted, and the number of lines processed. In a LINE comparison listing, as shown in Figure 196 on page 350, these are called *paired* changes. Paired changes indicate, for example, that a line inserted in the “new” data set corresponds to a line deleted from the “old” data set. The opposite would be true for non-paired insertions or deletions.

The number given as total changes should be the sum of the paired changes, non-paired insertions, and non-paired deletions. For example, in Figure 196 on page 350, the right-side totals are the equivalent of the following:

```
      2  PAIRED CHANGES
      1  NON-PAIRED INSERTS
+     0  NON-PAIRED DELETES
--
      3  TOTAL CHANGES
```

Bottom-of-Listing Information

The following list describes information that may appear at the bottom of your SuperC listings. Figure 198 on page 354 shows an example.

- The LISTING-TYPE generated: OVSUM, DELTA, CHNG, or LONG.
- The COMPARE-COLUMNS, or the range of columns compared, expressed as two numbers separated by a colon. For example, “1:76” means SuperC compared columns 1 to 76 in both data sets. If two column ranges are shown, the first is the range compared in the “new” data set or member and the second is the range compared in the “old” data set or member.
- The length in bytes of the LONGEST-LINE found in any data set or member compared.
- A list of the PROCESS OPTIONS USED.
- A list of the PROCESS STATEMENTS USED.
- Warnings and error messages.

Note: For BYTE comparisons, the COMPARE-COLUMNS information is not shown because column ranges are not applicable during BYTE comparisons.

Side-By-Side Listings

Two process options, NARROW and WIDE, let you produce listings that show source data side-by-side (Figure 199) instead of interspersed (Figure 196 on page 350). The NARROW process option shows up to 55 columns from each member, while the WIDE process option shows up to 80. The listing is truncated if it exceeds the line length of the process option being used.

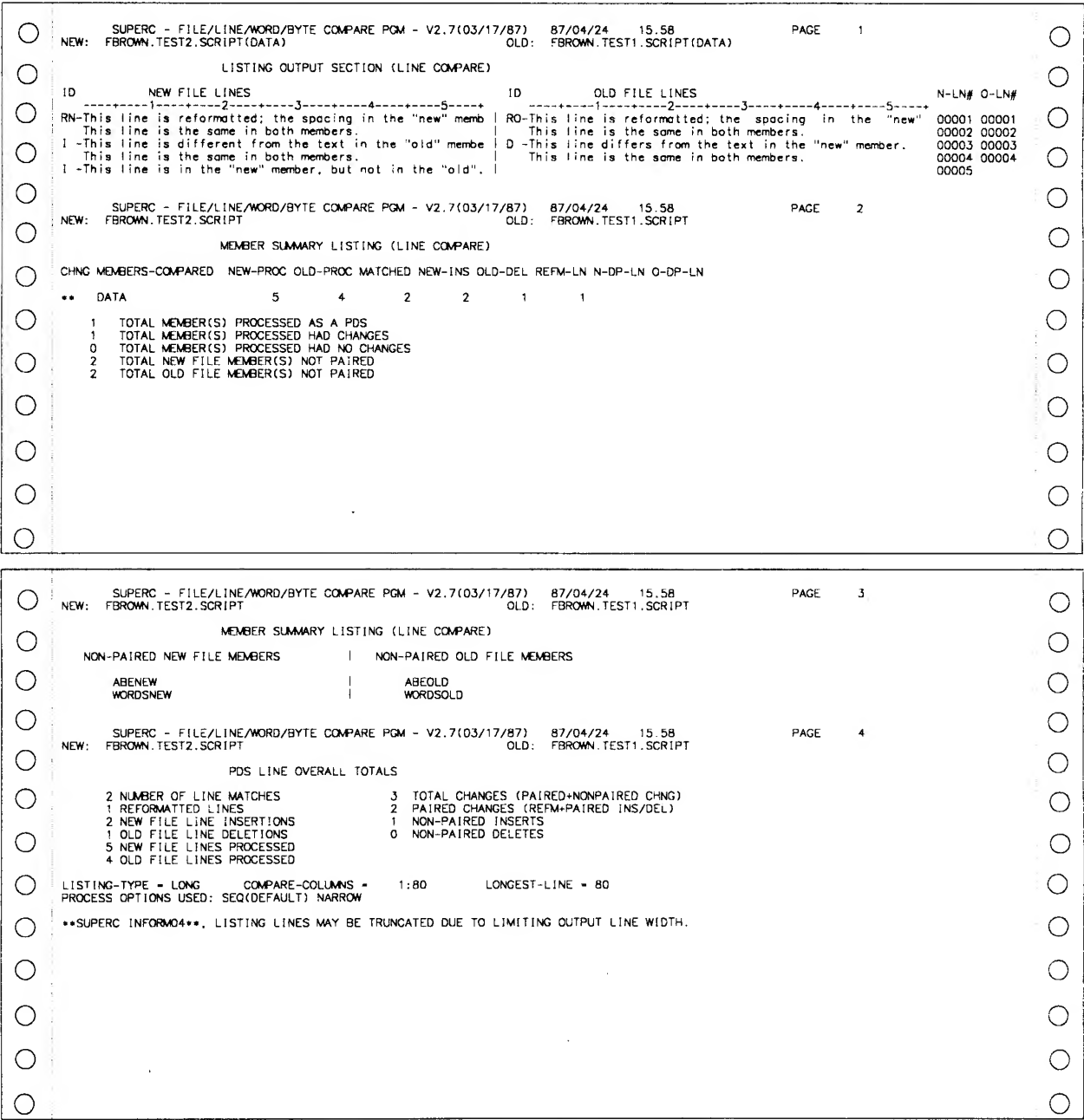


Figure 199. Side-By-Side LINE Comparison Using the NARROW Process Option

FILE Comparison of a Complete PDS

This type of comparison includes the Member Summary Listing section. As Figure 201 shows, the first information is the same as for a sequential/membered data set comparison, except the totals and hashsums are separated according to member name. Additional totals show how many members were:

- Processed and, of those, how many had changes
- Paired and how many were not paired.

The names of the paired and non-paired members are listed at the bottom.

```

SUPERC - FILE/LINE/WORD/BYTE COMPARE PGM - V2.7(87/02/13) 87/02/24 18.18
NEW: FBROWN.TEST2.SCRIPT OLD: FBROWN.TEST1.SCRIPT PAGE 1

MEMBER SUMMARY LISTING (FILE COMPARE)

DIFF SAME MEMBER-NAME N-BYTES O-BYTES N-LINES O-LINES N-HASH-SUM O-HASH-SUM
** DATA 400 400 5 5 8972446D 577EEEF8
MEMBER TOTALS 400 400 5 5

1 TOTAL MEMBER(S) PROCESSED AS A PDS
1 TOTAL MEMBER(S) PROCESSED HAD CHANGES
0 TOTAL MEMBER(S) PROCESSED HAD NO CHANGES
2 TOTAL NEW FILE MEMBER(S) NOT PAIRED
2 TOTAL OLD FILE MEMBER(S) NOT PAIRED

SUPERC - FILE/LINE/WORD/BYTE COMPARE PGM - V2.7(87/02/13) 87/02/24 18.18
NEW: FBROWN.TEST2.SCRIPT OLD: FBROWN.TEST1.SCRIPT PAGE 2

MEMBER SUMMARY LISTING (FILE COMPARE)

NON-PAIRED NEW FILE MEMBERS | NON-PAIRED OLD FILE MEMBERS
ABENEW | ABEOLD
WORDSNEW | WORDSOLD
PROCESS OPTIONS USED: APNDLST

```

Figure 201. FILE Comparison of a Complete PDS

Search-For Listings

```

SUPERC - FILE/LINE/WORD/BYTE COMPARE PGM - V2.7(87/02/13) 87/02/26 9.15 PAGE 1
LINE-# SOURCE SECTION SRCH DSN: FBROWN.TEST2.SCRIPT

DATA ----- STRING(S) FOUND -----

3 This line is the same in both members. 00030000
5 This line is the same in both members. 00050000

SUPERC - FILE/LINE/WORD/BYTE COMPARE PGM - V2.7(87/02/13) 87/02/26 9.15 PAGE 2
SUMMARY SECTION SRCH DSN: FBROWN.TEST2.SCRIPT

LINES-FOUND MEMBERS-W/LNS MEMBERS-PROC LINES-PROC COMPARE-COLS LONGEST-LINE
2 1 3 60 1: 80 80

THE FOLLOWING PROCESS STATEMENT(S) WERE PROCESSED:
SRCHFOR 'This line is'
SRCHFORC 'both members'

```

This listing was produced by searching all the members of a PDS named FBROWN.TEST.SCRIPT and by using the ASIS fields on the Extended Search-For Utility panel, as follows:

PAGE 1 of the listing shows that both strings were found in lines 3 and 5 of member DATA. However, instead of just showing the strings that were being searched for, the listing shows each line in its entirety.

- Two lines were found.
- One member of the PDS had lines that contained the search string.
- Three members of the PDS were searched and 60 of the lines those members contained were processed.
- In each member, columns 1 to 80 were searched.
- The longest line in any member was 80 bytes.

NW: 15457 DocId: 70002077 Page 754


automatically used the SRCHFOR process option because a search was requested. The SRCHFORC process option also was used because the "C" (continuation) keyword was specified at the end of the second ASIS field.

Appendix B. Character Translations for APL and TEXT Keyboards

You can use APL keyboards for all models of 3270 terminals, and TEXT keyboards for 3278 and 3279 terminals with PDF. Figure 203 and Figure 204 on page 364 show that the two-byte transmission codes for APL and TEXT characters are translated by PDF into one-byte codes for internal storage.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00																	0F
10																	1F
20																	2F
30																	3F
40	sp	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>E</u>	<u>F</u>	<u>G</u>	<u>H</u>	<u>I</u>	¢	.	<	(+		4F
50	&	<u>J</u>	<u>K</u>	<u>L</u>	<u>M</u>	<u>N</u>	<u>O</u>	<u>P</u>	<u>Q</u>	<u>R</u>	!	\$	*)	;	~	5F
60	-	/	<u>S</u>	<u>T</u>	<u>U</u>	<u>V</u>	<u>W</u>	<u>X</u>	<u>Y</u>	<u>Z</u>		,	%	-	>	?	6F
70		^	..						v	'	:	#	@	'	=	"	7F
80	~	a	b	c	d	e	f	g	h	i	↑	↓	≤	┌	└	→	8F
90	□	j	k	l	m	n	o	p	q	r	⌋	⌈		o		←	9F
A0	-	~	s	t	u	v	w	x	y	z	∩	U	↓	[≥	°	AF
B0	α	ε	ι	ρ	ω		×	\	÷		∇	Δ	τ]	≠		BF
C0	{	A	B	C	D	E	F	G	H	I	∞	∞		φ		⊙	CF
D0	}	J	K	L	M	N	O	P	Q	R	I	!	▼	▲	▢	▣	DF
E0	\		S	T	U	V	W	X	Y	Z	+	+		⊖	⊗	⊘	EF
F0	0	1	2	3	4	5	6	7	8	9		~	Δ	•	⊕		FF

 3278 only; invalid character on 3277.

 National use character. Graphics shown are for U.S. keyboards; graphics differ in other countries.

Character Translation for APL and TEXT Keyboards

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00																	0F
10																	1F
20																	2F
30																	3F
40	sp										¢	.	<	(+		4F
50	&	1	2	3						↓	!	\$	*)	;	¬	5F
60	-	/									!	,	%	_	>	?	6F
70	n	°								`	:	#	@	!	=	"	7F
80		a	b	ç	d	e	f	g	h	i	↑	{	≤	(+	+	8F
90	□	j	k	l	m	n	o	p	q	r		}	□)	±	■	9F
A0	—	~	s	t	u	v	w	x	y	z	●	L	Γ	[≥	•	AF
B0	0	1	2	3	4	5	6	7	8	9	▽	┐	└]	×	—	BF
C0	{	A	B	C	D	E	F	G	H	I	Δ	⊥	τ	v	←	→	CF
D0	}	J	K	L	M	N	O	P	Q	R	⊞	Δ	\$	¶	←	→	DF
E0	\	\	S	T	U	V	W	X	Y	Z	~	†	‡	♣	♥	^	EF
F0	0	1	2	3	4	5	6	7	8	9		L	Φ	Φ	Γ		FF

□ National use character. Graphics shown are for U.S. keyboards; graphics differ in other countries.

Figure 204. Internal Character Representations for TEXT Keyboards

Appendix C. List of Abbreviations

The following list includes the command names, field values, keywords/operands, and scroll amounts that can be abbreviated, followed by the allowable abbreviations. To improve readability, we recommend that you avoid using abbreviations in edit macros.

Command Name Abbreviations

BACK	B		
BROWSE	BRO		
CANCEL	CAN		
COLUMNS	COLS	COL	
CONFIRM	CON	C	
DISPLAY	DISPL	DISP	DIS
FIND	F		
INDEX	I		
LOCATE	LOC	L	
RESUME	RES		
SELECT	SEL	S	
SHOWCMD	SHOW		
SKIP	S		
TOP	T		
UP	U		

Field Value Abbreviations

%USED	%		
AFTER	AFT	A	
ALIAS-OF	ALIAS		
ATTRIBUTES	ATTR		
BEFORE	BEF	B	
BLOCK	BLKS		
BOTTOM	BOT	B	
CHANGED	CHA	CHG	C
CREATED	CRE		
CYLINDER	CYLS		
DEVICE	DEV		
EXPIRES	EXP		
GET	G		
MESSAGE	MES		
NO	N		

Keyword/Operand and Scroll Amount Abbreviations

PUT	P	
REFERRED	REF	
TOP	T	0
TRACK	TRKS	
VOLUME	VOL	
YES	Y	

Keyword/Operand Abbreviations

CHANGE	CHA	CHG	C
CHARS	CHAR		
COBOL	COB		
DISPLAY	DISPL	DIS	
ERROR	ERR		
LABELS	LABEL	LAB	
PREFIX	PRE		
SPECIAL	SPE		
STANDARD	STD		
SUFFIX	SUF		
VERTICAL	VERT		
VOLUME	VOL		

Scroll Amount Abbreviations

CUR	CSR
DATA	D
HALF	H
MAX	M
PAGE	P

Programming Language Abbreviations

ASSEMBLER	ASSEMBLE	ASSEM	ASM
COBOL	COB		
FORTRAN	FORT	FOR	FTN
PASCAL	PAS		
PLI	PLIOPT	PL1	
SCRIPT	SCR	TEXT	

Appendix D. Allocation Data Sets

ISPF issues ALLOC commands based on the ISPF libraries, data set names, list IDs, options, and additional input libraries you enter on the Foreground Assembler H and Foreground VS COBOL II Compile panels. All allocations are done before Assembler H and the VS COBOL II compiler are invoked.

Since Assembler H and VS COBOL II do not provide a language prompter, PDF allocates the required data sets for you. The following sections describe the data sets PDF allocates when you use Assembler H or the VS COBOL II compiler.

SYSIN Data Set

The SYSIN data set is the main input into Assembler H and VS COBOL II. It contains the ISPF libraries or “other” partitioned or sequential data sets that you enter on the Foreground Assembler H and VS COBOL II Compile panels. This data set is used to find the member that contains the program to be assembled or compiled. For a PDS, the ALLOC command would be:

```
ALLOC FI(SYSIN) DA('proj.lib.type(mem)')
```

For a sequential data set, the ALLOC command would be:

```
ALLOC FI(SYSIN) DA('proj.lib.type')
```

In both commands, “lib” is the library in which the member or data set was found.

SYSLIB Data Set

The SYSLIB data set contains the ISPF library concatenation sequence used to resolve any copy statements specified in your program. It contains the ISPF libraries or “other” partitioned or sequential data sets and the additional input libraries you specify on the Foreground Assembler H and VS COBOL II Compile panels. For example:

```
ALLOC FI(SYSLIB) DA('SYS1.MACLIB','proj.lib1.type',...,  
                  'proj.lib4.type','additional lib1','additional lib2')
```

SYSPRINT, SYSTEM, and SYSLIN Data Sets

SYSPRINT Data Set

The SYSPRINT data set contains the generated output listing. The entry in the LIST ID field determines the destination of the output listing. If you enter a name in the LIST ID field, the output listing is stored in a sequential data set:

```
ALLOC FI(SYSPRINT) DA('prefix.listid.LIST')
```

where “listid” is the name entered in the LIST ID field. However, if you leave the LIST ID field blank, PDF uses the name of the member being assembled or compiled instead of the list ID:

```
ALLOC FI(SYSPRINT) DA('prefix.member.LIST')
```

If you enter an asterisk (*) in the LIST ID field, ISPF displays the output listing at your terminal, using the following command:

```
ALLOC FI(SYSPRINT) DA(*)
```

See “List Data Sets” on page 216 if you need more information.

SYSTEM Data Set

The SYSTEM data set contains a summary of the information in the listing data set (SYSPRINT). It is displayed at the terminal if the TERM option is used:

```
ALLOC FI(SYSTEM) DA(*)
```

SYSLIN Data Set

The SYSLIN data set contains the object module. This object module will be the input when you link edit. For a PDS, the ALLOC command would be:

```
ALLOC FI(SYSLIN) DA('proj.lib1.OBJ(mem)')
```

For a sequential data set, the ALLOC command would be:

```
ALLOC FI(SYSLIN) DA('proj.lib1.OBJ')
```


SYSPUNCH, SYSUT1, and SYSUT2-SYSUT5 Data Sets

SYSPUNCH Data Set

The SYSPUNCH data set is the same as the SYSLIN data set. ISPF does not use this data set. The DUMMY parameter on the ALLOC statement means it should not be used:

```
ALLOC FI(SYSPUNCH) DUMMY
```

SYSUT1 Data Set

The SYSUT1 data set is a temporary utility data set used during processing. It is deleted after it is used.

For Assembler H, the format is:

```
ALLOC FI(SYSUT1) UNIT(SYSDA) NEW DELETE
```

For VS COBOL II, the format is:

```
ALLOC FI(SYSUT1) UNIT(SYSDA) NEW DELETE SPACE(1,1) CYLINDER
```

SYSUT2 - SYSUT5 Data Sets

The SYSUT2, SYSUT3, . . . ,SYSUT5 data sets are temporary utility data sets used by VS COBOL II *only* during processing. They are deleted after they are used:

```
ALLOC FI(SYSUT2) UNIT(SYSDA) NEW DELETE SPACE(1,1) CYLINDER
ALLOC FI(SYSUT3) UNIT(SYSDA) NEW DELETE SPACE(1,1) CYLINDER
ALLOC FI(SYSUT4) UNIT(SYSDA) NEW DELETE SPACE(1,1) CYLINDER
ALLOC FI(SYSUT5) UNIT(SYSDA) NEW DELETE SPACE(1,1) CYLINDER
```

SYSPUNCH, SYSUT1, and SYSUT2-SYSUT5 Data Sets

Appendix E. SuperC Reference

PDF contains two utilities, SuperC (option 3.12) and SuperCE (option 3.13), that allow you to compare data sets for differences. Also, PDF contains two other utilities, Search-For (option 3.14) and Extended Search-For (option 3.13.S), that allow you to search data sets for strings of data.

All four of these utilities combine two major components to do their respective functions. The first component is a dialog that provides the data entry panels, selection panels, and messages.

The second component is the SuperC program, which is a major rewrite of the original IBM Source Compare/Audit Utility (Installed User Program #5796 PLZ, Copyright 1979). It is upwardly compatible with the old MVS IUP version and can serve as its direct replacement. All four utilities use a standardized parameter list to invoke the SuperC program.

You can use the SuperC program without the PDF utilities. To do this, however, your installation must customize a CLIST (for interactive use) or a PROCLIB procedure (for batch execution of a catalog procedure). A sample CLIST has been provided to show "line" command invocation. A sample PROCLIB JCL catalog procedure has also been provided to illustrate batch submission.

Utility Differences

The standard utilities, SuperC (option 3.12) and Search-For (option 3.14), can be characterized as easy to use with somewhat reduced function. On the other hand, the extended utilities, SuperCE (option 3.13) and Extended Search-For (option 3.13.S), fully exploit the SuperC program's capabilities.

Standard Utilities

The standard utilities are useful for ordinary comparisons and searches. The SuperC utility (option 3.12) uses a two-panel sequence: you specify the "new" input data set on the first panel and the "old" input data set on the second. The Search-For utility (option 3.14) uses an optional two-panel sequence: you can specify the input data set and one search string on the first panel, or use the second panel if you need to specify more than one search string.

You can enter additional information on these panels as they are displayed. If you are using the SuperC utility, you can enter the name of a previously prepared profile data set that contains additional information to specify the comparison.

SuperC - Program Description

Search-For does not use a profile data set. Also, Search-For finds all occurrences without case distinction when searching for a data string.

Extended Utilities

The primary intent of the extended utilities is to provide maximum flexibility and access to all SuperC functions. Input fields are provided to let you use process options and statements. Also, the Extended Search-For utility's ASIS fields let you specify mixed case search strings.

The input data set name fields differ from standard PDF format because PROJECT, GROUP, TYPE, and MEMBER fields are not provided. Instead, you can enter input data set names horizontally using standard TSO naming conventions. This includes the use of a PDS member name, if desired, as part of a data set name.

The concatenation of input data sets is also different. Up to four data set names, as opposed to the standard four ISPF library groups, can be entered as "new" or "old" data sets. This allows data sets with the same attributes to be concatenated. For example, PANELS and MSGS data sets could be concatenated for searching.

Besides compare functions, the SuperCE Utility panel provides access to the Extended Search-For utility (option 3.13.S). This gives you the added advantage, besides access to more functions than Search-For (option 3.14) provides, of the ability to search for a data string without having to leave SuperCE.

Program Description

The SuperC program is a fast and versatile program that can process:

- Two sequential data sets
- Two complete partitioned data sets
- Members of two partitioned data sets
- Concatenated data sets.

In fact, any data set that can be processed by the normal OS sequential access method can be processed by the SuperC program. SuperC can compare data sets even when there are many differences and much redundant data. Some examples of redundant data are blank lines, duplicate words, and binary data with many duplicate characters.

Unlike many compare programs, SuperC is not limited to comparing data sets on a line-by-line basis. Instead, it lets you choose between the following four comparison levels. The compare type you select determines which kinds of data differences are presented by SuperC. See "Comparison Result Differences" on page 374 for more information about comparison results.

- FILE comparisons produce summary information about the differences between the data sets being compared.

- LINE comparisons are record-oriented and note matching, inserted, deleted, and reformatted lines. This level is most useful for comparing lines of program source code. It provides the least output difference information and is least sensitive to resynchronization.
- WORD comparisons note differences between data strings that are delimited by blanks or non-alphanumeric characters, such as commas. Matching words are found, even if they are not on the same line. This level is most useful for comparing text data sets.
- BYTE compare determines byte differences. It is most useful for comparing unformatted and machine-readable data.

The SuperC *program* requires only the names of the input data sets. However, the *utility* you are using may require other information, such as a listing type. Also, you can enter the following types of processing information and options on the utility data entry panels:

- Compare type
- Listing data set name or destination
- Process options
- Statements or profile data set name
- Browse output designation.

The SuperC program lets you create two kinds of output: 1) a listing that shows the results of the comparison or search, and 2) a structured data set that contains update information. Within these two categories, you can create many kinds of output that make it easy to see where your data differs. To see your comparison results, you can generate listings that show:

- An overall summary of total changes
- The actual source code where deltas (differences) were found
- The deltas plus up to 10 matching lines before and after
- The deltas plus all matching lines.

You also have a choice between listings that show differences either sequentially or side-by-side. “SuperC Listings” on page 348 explains the listing sections and provides examples. Also, see “Search-For Listings” on page 360 for information about Search-For listings.

In an update data set, output lines are identified and results are put in specific columns. An update data set is especially useful as input to a user-written application program. It allows a program to customize what you see, changing generalized output to information that is specific to a particular application. “Update Data Set Control Options” on page 385 provides explanations and examples of the various update data sets you can produce.

SuperC - Program Description

SuperC operates independent of any synchronization data, such as column or sequence numbers. It does not use the common “start at the top,” then look ahead or look back method to determine large sections of matching data. Neither does it sort the data before comparing data units. Instead, SuperC recognizes matching and missing ⁵ files, lines, words, or bytes (data units) based on data content only. It finds all matches, locates the largest set of matching data units, and recursively allows this compare set to divide the data set into three smaller subsections. All sections are processed for matches. The process ends when no more matches can be found within the partitioned subsections. Sections classified as inserted or deleted are sections for which no matches were found.

The SuperC program does not unpack data before doing a comparison or conducting a search. Therefore, comparisons of packed to unpacked data may not produce significant results. You can successfully compare packed data to packed data. However, the listing will show any differences in packed format.

The SuperC utilities (options 3.12, 3.13, and 3.14) do not explicitly support formatted or DBCS data. You can successfully compare and search data sets containing DBCS data, but precautions must be taken to avoid splitting DBCS characters in the output listing. It is recommended that only the LINE compare type be used and that the LSTCOLM process statement be avoided.

Comparison Result Differences

Different compare types often give you slightly different results. Here are some of the reasons:

- FILE and BYTE comparisons inspect the complete data set (and every byte) for differences. LINE and WORD comparisons use designated columns that are either your specifications or SuperC defaults. These two compare types may use different columns, especially when you specify that sequence numbers are to be ignored.
- LINE comparisons pad variable-length records with blanks when lines with different lengths are compared. However, BYTE comparisons can find blank characters only if they are included in the input data sets.
- LINE comparisons of normal input source lines compress blanks from the input lines before developing matched sets. However, input lines longer than 256 characters use the complete input line. This difference could produce unexpected results.
- Data sets may have different sensitivity to being resynchronized. The largest matching compare set between the data sets being compared divides the input data. LINE compare units, WORD compare units, or BYTE compare units may be different sizes even though the input data is identical.

- LINE comparisons may produce different results because of lines being reformatted. However, reformatted lines have no effect on WORD comparisons because they ignore blank characters and blank lines.
- Results may differ depending on which compared data set is specified as “new” and which is specified as “old.” The matching algorithm may be sensitive to the first matched set it finds between data sets.

Applications

You can use the SuperC program for many applications other than the traditional “compare two source data sets.” This section lists some specific applications for general users, writers and editors, and programmers and systems administrators.

General users can:

- Compare two data sets that have been reformatted. Reformatted data sets contain such differences as indentation level changes, spaces inserted or deleted, or lines that have been reformatted and moved to other parts of the data set.

SuperC detects and classifies reformatted lines as special changes. You can list these lines in the output, along with the normal insert/delete changes, or eliminate them from the listing. Reducing the number of flagged lines may help you focus on real, rather than cosmetic, changes.

- Determine whether two PDSs, or a concatenation of PDSs, have corresponding like-named members.

Members absent from one data set but present in the other are listed, besides all change activity between like-named members. This may show changes caused by creating or deleting PDS members.

Writers and editors can:

- Detect word changes within documents.

SuperC finds word differences even if the words have been moved to different lines.

- Verify that only designated areas are changed.

SuperC comparison results show all areas affected. Changes made to restricted areas may be invalid. Therefore, unintended changes can be detected so that a complete document need not be checked for errors again.

- Create a utility that automatically inserts SCRIPT revision codes.

You could write a program that uses WORD compare to find where words in the “new” data set are different, makes a copy of the “new” data set, and then inserts SCRIPT revision codes (.RC) before and after

the changed words. This utility could eliminate the need to insert SCRIPT revision codes manually.

Programmers and systems administrators can:

- Generate management reports that show the quantity and type of changes in program source code.

SuperC can count the changed and unchanged lines of code in an application program. Therefore, comparison results could be used to summarize the changes between different versions of a program.

- Retain a record of change activity.

Listing data sets can be collected and retained as a permanent record of the changes made before a new program is released. Source code differences can help detect regressions or validate the appropriateness of any code modifications.

- Rewrite a listing data set, including additional headers or change delimiters.

Some SuperC listings may need to be rewritten before you accept the results. For example, some installations may require security classifications. Others may require a listing created using the WIDE process option to have box delimiters surrounding changed sections.

- Compare data sets across non-connected systems.

SuperC can generate a 32-bit hashsum per data set or member using the FILE compare type. Data sets compared on a non-connected processor, using SuperC, should have the same hashsums if they are identical. A FILE comparison of any data set to determine a hashsum can be done by specifying the same data set as both "new" and "old."

- Develop additional uses for update data sets.

SuperC produces general results with generalized reports. However, your installation may have additional requirements. There are many specialized update formats that you can use to produce listings that match these requirements. Normal SuperC listings may not fit this type of application, but the update data sets are more structured and should be easier to use as data input. See "Update Data Set Control Options" on page 384 for explanations and examples of the update data sets.

Return Codes

Results from invoking the SuperC program are returned in register 15. The SuperC and SuperCE utilities generate a short message and a long message to interpret the results for you. However, only the long message displays the return codes, which are:

- | | |
|----|--|
| 0 | Normal completion. No differences were found in the input data sets. |
| 1 | Normal completion. Differences were found in the input data sets. |
| 4 | Warning. Erroneous/conflicting input options were detected by the SuperC program. Data sets were compared, but should be checked for results consistent with those expected. Check listing for more details. |
| 6 | Warning. The old data set did not contain proper sequence numbers, or the sequence number intervals were not large enough to contain insert activity. This return code applies only to results from the UPDCMS8, UPDMVS8, and UPDSEQ0 process options. |
| 8 | Error. Error on "old" input data set. The data sets were <i>not</i> compared. Check output for more information. |
| 12 | Error. Inconsistent data set organizations. The "new" and "old" data sets are not both PDSs or sequential data sets. The compare was not run. |
| 16 | Error. Error on "new" input data set. The data sets were <i>not</i> compared. Check output for more information. |
| 20 | Warning. Error on update data set; one of the following: <ul style="list-style-type: none"> • Missing DELDD • I/O error • Append to PDS • Incorrect attributes on an append or PDS output operation. |

Processing was completed with the update options reset and ignored. The error can also be returned if an update data set is being used simultaneously as the listing data set.

- | | |
|----|--|
| 24 | Error. Error occurred during open or during writing to the listing data set. The error can also be returned because of a "new" or "old" input data set simultaneously being used as the listing data set. No additional information is generated to better define the error condition. |
| 25 | Error. The listing data set was not allocated because of a conflict with the logical record length (LRECL), record format (RECFM), or data set organization. A listing can be saved only in a PDS that has the same LRECL or RECFM; a listing can never be |

SuperC - Process Options

appended to a PDS by using the APNDLST process option; a listing can be appended by using the APNDLST process option only to a sequential data set that has the same LRECL or RECFM. No output was generated.

- 26 Error. The listing data set could not be extended and is full (E37 condition). The results are truncated and incomplete.
- 28 Error. No data was compared because the “old” and “new” data sets did not have members with the same name or because both input data sets were empty. See the listing for more information.
- 32 Error. Insufficient storage was available for the SuperC program to run. The user region size was either too small or storage was too fragmented.

Process Options

Option P on either the SuperCE Utility panel or the Extended Search-For Utility panel provides one or more panels from which you can select process options. For SuperCE, the options displayed are compatible with the compare type (FILE, LINE, WORD, or BYTE) that you specified in the Compare Type field.

In the following tables, all process options that are identified as **Options** are valid as both **Compare Options** and **Search Options**, as explained below.

All process options that are identified as **Compare Options** can be:

- Used in a profile data set with the SuperC utility (option 3.12).
- Selected from a panel or entered in the Process Options field on the SuperCE Utility panel (option 3.13).

All process options that are identified as **Search Options** can be selected from a panel or entered in the Process Options field on the Extended Search-For Utility panel (option 3.13.S). However, they are *not* available for the Search-For utility (option 3.14).

Input Process Control Options

The first table defines process options that determine which parts of the input data sets are processed.

Options	Definitions	Compare Types
ANYC	Any case. Converts data to be compared or searched, and search strings, to uppercase before processing.	LINE and WORD
COBOL	Ignore columns 1-6 in fixed block 80-byte record length (FB 80) data sets during a search or comparison. Data in columns 1-6 is assumed to be sequence numbers. Cannot be used with NOSEQ or SEQ.	LINE and WORD
NOSEQ	Search or compare complete data set line, regardless of record type. That is, treat standard sequence number columns as data during a search or compare. Sequence numbers are assumed in columns 73-80 in FB 80 and columns 1-8 in VB 255. This is the default condition when you are using the Extended Search-For utility (option 3.13.S). Cannot be used with COBOL or SEQ.	LINE and WORD
SEQ	Ignore standard sequence numbers in FB 80 and VB 255 records. That is, do not search or compare those columns. Sequence numbers are assumed in columns 73-80 in FB 80 and columns 1-8 in VB 255. This is the default condition when you are using the SuperC utility (option 3.12) or the SuperCE utility (option 3.13) when you are comparing FB 80/VB 255 data sets. Cannot be used with COBOL or NOSEQ.	LINE and WORD

Figure 205. Input Process Control Options

SuperC - Process Options

"Don't Process" Control Options

The next table defines process options that can prevent parts of the input data sets from being processed. For comparisons, these process options are available for the LINE and WORD compare types only. Also, for a WORD comparison and for searches, you must type them in the Process Options field, because they are not available on selection panels.

Options	Definitions
DPACMT	Don't process asterisk (*) comment lines. Assembler comment lines with "*" in column 1 are excluded from the compare set.
DPADCMT	Don't process ADA-type comments. ADA comments, such as "-- comment ...," and blank compare lines are excluded from the compare set.
DPBLKCL	Don't process blank compare lines. Source lines in which all the compare columns are blank are excluded from the compare set.
DPCBCMT	Don't process COBOL-type comment lines. COBOL source lines with an "*" comment character in column 7 are excluded from the compare set.
DPFTCMT	Don't process FORTRAN-type comment lines. FORTRAN source lines with a "C" comment character in column 1 are excluded from the compare set.
DPPLCMT	Don't process PL/I-type comments. PL/I, C, and REXX comments, such as /* . . . */, and blank compare lines are excluded from the compare set. <i>Note: Conflicts between quoted strings and comment delimiters occasionally cause incorrect comment recognition.</i>
DPPSCMT	Don't process Pascal-type comments. Pascal comments, such as (* . . . *), and blank compare lines are excluded from the compare set. Both DPPSCMT and DPPLCMT may be required for some Pascal compilers.

Figure 206. "Don't Process" Control Options

Output Process Control Options

The next table defines process options that affect the format and content of the output listing data set. See "SuperC Listings" on page 348 for examples of SuperC listings and "Search-For Listings" on page 360 for an example of a Search-For listing.

Compare Options	Search Options	Definitions	Compare Types
APNDLST	APNDLST	Append the incremental results of the current compare to an existing listing report that has the same data set attributes. This lets you collect multiple listing results under one data set name.	FILE, LINE, WORD, and BYTE
CNPML	N/A	Count non-paired PDS member lines for group summary. Use this option to inventory the total number of processed and unprocessed lines. Otherwise, only paired entries are listed with line counts.	LINE
DLMDUP	N/A	Don't list matching duplicate lines. "Old" data set source lines that match "new" data set source lines are omitted from a side-by-side (NARROW or WIDE) output listing.	LINE
DLREFM	N/A	Don't list reformatted lines. "Old" data set source lines that are the same except for the number of blanks and location of blanks are omitted. Only "new" data set reformatted lines are included in the listing.	LINE
FMVLNS	N/A	Flag moved lines. "Moved" lines are flagged as Insert-Moved (IM) in the "new" data set and Delete-Moved (DM) in the "old" data set. Reformatted moved lines are specifically identified in the "TYPE" listing column.	LINE
GWCBL	N/A	Generates WORD compare change listings. Lists the "new" data set lines, highlighting the changed lines (words can be inserted or deleted from the line) with change bar flagging.	WORD
N/A	IDPFX	Identifier prefixed. A member name appears in the output listing at the beginning of each line shown, so you can tell immediately the member in which the line was found.	N/A
N/A	LMT0	List member totals. Lists the member summary totals and the overall summary totals of the PDS search-for comparison. Cannot be used with LPSF or LTO.	N/A
LOCS	N/A	List only changed entries in an OVSUM listing. Unchanged paired entries and non-paired "new" and "old" data set entries are omitted.	FILE, LINE, WORD, and BYTE
LONGLN	LONGLN	Long lines. Lists up to 176 columns from the input data sets instead of the usual 106 columns. Cannot be used with NARROW or WIDE.	LINE

Figure 207 (Part 1 of 2). Output Process Control Options

SuperC - Process Options

Compare Options	Search Options	Definitions	Compare Types
N/A	LPSF	List previous search string and following lines. Lists the matched string line and up to six preceding and six following lines for context. Cannot be used with LMTO or LTO.	N/A
N/A	LTO	List totals only. Lists the general summary total of the search results. Cannot be used with LMTO or LPSF.	N/A
NARROW	N/A	Creates a listing data set with 55 columns from each source data set line. Insertions and deletions are flagged and appear side-by-side in the listing output. The listing is usually 133 columns wide. However, if the NOPRTCC option is used, the listing is only 132 columns wide because the printer carriage control characters are not included. Cannot be used with LONGLN or WIDE.	LINE
NOPRTCC	NOPRTCC	No printer control column. The listing data set omits the normal listing control column and page separators. Section separators and title lines are still generated. This data set may be better for browsing at a terminal.	FILE, LINE, WORD, and BYTE
REFMOVR	N/A	Reformat override. Prevents reformatted lines from being flagged in the listing. They are, however, counted for the overall summary statistics.	LINE
WIDE	N/A	Creates a listing data set with 80 columns from each source data set line. Insertions and deletions are flagged and appear side-by-side in the listing output. The listing is usually 203 columns wide. However, if the NOPRTCC option is used, the listing is only 202 columns wide because the printer carriage control characters are not included. Cannot be used with LONGLN or NARROW.	LINE
XWDCMP	N/A	Extended word compare. The word delimiter set is extended to include all non-alphanumeric characters, including blanks. The XWDCMP option causes SuperC to see these characters and blanks as words, and word delimiters. For example, " ABCD(EFGH) JKL " would be two words using normal WORD compare, but is considered to be five words (three words and two word delimiters) when you use the XWDCMP option.	WORD

Figure 207 (Part 2 of 2). Output Process Control Options

Update Data Set Control Options

This section describes the process options that are used to create update data sets. You cannot enter two different process options that begin with UPD. These options are compare options only.

Each description except APNUPD is accompanied by at least one sample update data set. All the samples were created by comparing member DATA, a PDS member common to both FBROWN.TEST1.SCRIPT and FBROWN.TEST2.SCRIPT. Figure 195 on page 349 shows the contents of this member in both PDSs.

APNDUPD - Append Update Data Set

This process option appends the incremental results of the current compare to an existing update report. It lets you collect multiple update results under one data set name. However, the width of the update data set being appended must be the same as the width of the original listing. Therefore, update options (UPDxxxx) that would change the width cannot be used with APNDUPD unless the original listing was produced by the same option(s).

The APNDUPD process option is available with the LINE, WORD, and BYTE compare types.

UPDCMS8 - Update CMS Sequenced 8

This process option produces a data set that contains both control statements and "new" data set source lines using sequence numbers from "old" data set columns 73-80. It is available with the LINE compare type only.

All UPDCMS8 control statements are identified by the characters "./" in columns 1 and 2 of the 80-byte record, followed by one or more blanks and additional blank-delimited fields. The control statements are sequence (S), insert (I), delete (D), replace (R), and comment (*).

Data sets created with the UPDCMS8 process option can be used as input to the CMS UPDATE command, described in *CMS Command and Macro Reference*. Figure 208 shows an example of a UPDCMS8 update data set.

```

BROWSE -- FBROWN.UPDATE.CMS8(LINE) ----- LINE 00000000 COL 001 080
COMMAND ==> _                               SCROLL ==> PAGE
***** TOP OF DATA *****
./ * NEW: FBROWN.TEST2.SCRIPT(DATA)           87/02/24 17.34
./ * OLD: FBROWN.TEST1.SCRIPT(DATA)
./ R 00010000 00010000 $ 00012990 00002990
This line is in the "new" member, but not in the "old".           00010000
This line is reformatted; the spacing in the "new" member differs. 00020000
./ R 00030000 00030000 $ 00034990 00004990
This line differs from the text in the "old" member.             00040000
./ D 00050000 00050000
***** BOTTOM OF DATA *****

```

Figure 208. UPDCMS8 Update Data Set

The record attributes for both input data sets must be FB 80. They must also contain sequence numbers.

SuperC - Process Options

UPDCNTL - Update Control

This process option produces a control data set that relates matches, insertions, deletions, and reformats to:

- The relative line numbers of the “old” and “new” data sets (LINE compare)
- The relative word position of the “old” data set (WORD compare)
- The relative byte offset (BYTE compare).

No source or data from either input data set is included. The following panels show the format used with each of these compare types.

UPDCNTL Data Set Using LINE Compare

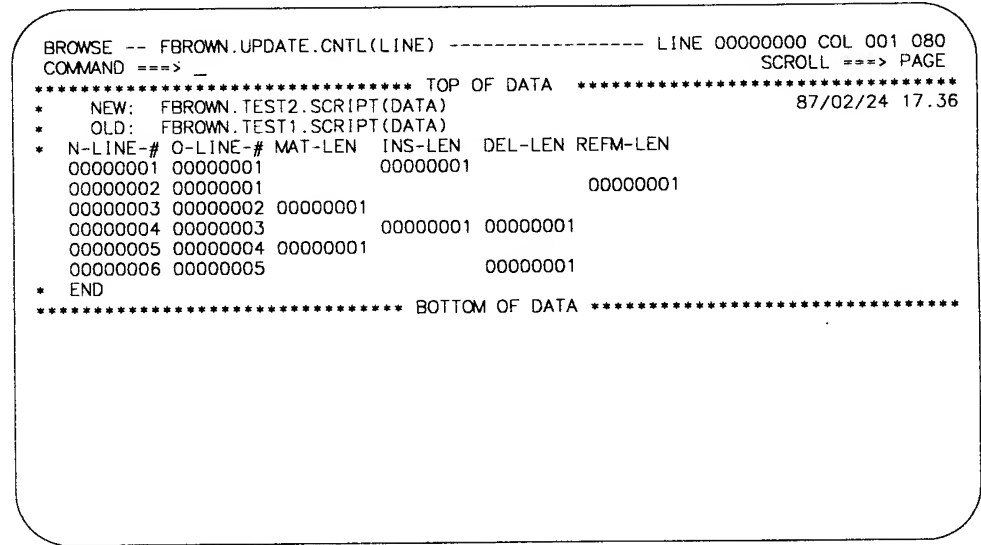


Figure 209. UPDCNTL Data Set Using LINE Compare

The top of Figure 209 shows the names of the “new” and “old” input data sets, along with the date and time the update data set was created. The following table shows the column numbers used to display the update information:

Column Numbers	Data Item
4-11	New line number
13-20	Old line number
22-29	Match length
31-38	Insert length
40-47	Delete length
49-56	Reformat length

Figure 210. UPDCNTL Format Using LINE Compare

UPDCNTL Data Set Using WORD Compare

```

BROWSE -- FBROWN.UPDATE.CNTL(WORD) ----- LINE 00000000 COL 001 080
COMMAND ==> -                               SCROLL ==> PAGE
***** TOP OF DATA *****
* NEW: FBROWN.TEST2.SCRIPT(DATA)
* OLD: FBROWN.TEST1.SCRIPT(DATA)
* N-LINE-# N-LN-LEN N-COL WD-MAT-# N-WD-INS O-WD-DEL O-LINE-# O-LN-LEN O-COL
00000001 00000001 00001 00000010 00000013
00000002 00000001 00001 00000010 00000002 00000001 00000001 00000001 00001
00000003 00000001 00001 00000008 00000001 00000001 00000002 00000001 00001
00000004 00000001 00001 00000008 00000001 00000001 00000002 00000001 00040
00000005 00000001 00001 00000008 00000001 00000001 00000003 00000001 00001
00000006 00000001 00001 00000008 00000001 00000001 00000003 00000001 00040
00000007 00000001 00001 00000008 00000001 00000001 00000003 00000001 00046
00000008 00000001 00001 00000008 00000001 00000001 00000003 00000001 00054
00000009 00000001 00001 00000008 00000001 00000001 00000004 00000001 00001
00000010 00000001 00001 00000008 00000001 00000001 00000004 00000001 00040
00000011 00000001 00001 00000008 00000001 00000001 00000005 00000001 00057
* END
***** BOTTOM OF DATA *****

```

Figure 211. UPDCNTL Listing Using WORD Compare

The top of Figure 211 shows the names of the “new” and “old” input data sets, along with the date and time the update data set was created. The following table shows the column numbers used to display the update information:

Column Numbers	Data Item
4-11	Beginning new line number
13-20	Number of lines
22-26	New column number (beginning of word)
28-35	Number of matching words
37-44	Number of new inserted words
46-53	Number of old deleted words
55-62	Beginning old line number
64-71	Number of old lines
73-77	Old column number

Figure 212. UPDCNTL Format Using WORD Compare

SuperC - Process Options

UPDCNTL Data Set Using BYTE Compare

```
BROWSE -- FBROWN.UPDATE.CNTL(BYTE) ----- LINE 00000000 COL 001 080
COMMAND ==> - SCROLL ==> PAGE
***** TOP OF DATA *****
* NEW: FBROWN.TEST2.SCRIPT(DATA)
* OLD: FBROWN.TEST1.SCRIPT(DATA)
* N-BYTE-O O-BYTE-O MAT-LEN INS-LEN DEL-LEN
00000000 00000000 00000050
00000050 00000000 0000001E
0000006E 0000001E 00000001
0000006E 0000001F 00000008
00000076 00000027 00000001
00000076 00000028 00000002
00000078 0000002A 00000001
00000078 0000002B 00000004
0000007C 0000002F 00000001
0000007C 00000030 00000007
00000083 00000037 00000001
00000083 00000038 00000006
00000089 0000003E 00000001
00000089 0000003F 00000009
00000092 00000048 00000006
00000098 00000048 00000003
00000098 0000004B 00000001 00000001
0000009C 0000004C 0000004F
```

```
BROWSE -- FBROWN.UPDATE.CNTL(BYTE) ----- LINE 00000022 COL 001 080
COMMAND ==> - SCROLL ==> PAGE
000000EB 0000009B 00000001 00000001
000000EC 0000009C 0000002C
00000118 000000C8 00000003 00000003
0000011B 000000CB 00000020
0000013B 000000EB 00000001 00000001
0000013C 000000EC 0000004F
0000018B 0000013B 00000050
0000018B 0000018B 00000005
* END
***** BOTTOM OF DATA *****
```

Figure 213. UPDCNTL Listing Using BYTE Compare

The top of the first panel in Figure 213 shows the names of the “new” and “old” input data sets, along with the date and time the update data set was created. The second panel shows the result of scrolling down to see the remainder of the data set.

The following table shows the column numbers used to display the update information:

Column Numbers	Data Item
4-11	New byte offset
13-20	Old byte offset
22-29	Number of matching bytes
31-38	Number of inserted bytes
40-47	Number of deleted bytes

Figure 214. UPDCNTL Format Using BYTE Compare

UPDMVS8 - Update MVS Sequenced 8

This process option produces a data set that contains both control statements and "new" data set source lines using sequence numbers from "old" data set columns 73-80. This process option is available for the LINE compare type only.

The format of the generated data may be suitable as input to the IEBUPDTE utility. See *OS/VS2 MVS Utilities* for information about the contents of this data set. Figure 215 shows an example of an UPDMVS8 update data set.

```

BROWSE -- FBROWN.UPDATE.MVS8(LINE) ----- LINE 00000000 COL 001 080
COMMAND ==> _                                SCROLL ==> PAGE
***** TOP OF DATA *****
./ CHANGE LIST=ALL,NAME=DATA      OLD:FBROWN.TEST1.SCRIPT
This line is in the "new" member, but not in the "old".                00010000
This line is reformatted; the spacing in the "new" member differs.      00015000
This line differs from the text in the "old" member.                    00030000
./ DELETE  SEQ1=00050000,SEQ2=00050000
***** BOTTOM OF DATA *****

```

Figure 215. UPDMVS8 Update Data Set Listing

The data sets to be compared must have a record format of FB 80. They must also contain sequence numbers.

SuperC - Process Options

UPDPDEL - Prefixed Delta Lines

This process option produces a data set that contains header records and complete delta lines from the input source data set(s), up to a maximum of 32K columns in each output line. It is available for the LINE compare type only.

The data set produced is a variable length data set that reflects the input source data set(s)'s characteristics. Prefix codes, such as "I" for insert, and the line number precede lines from the source data sets. Abbreviations show the total number of consecutive lines flagged as changed, such as "INS#=" for the number of consecutive inserted lines. The number of replaced (RPL#=) lines always refers to consecutive *pairs* of inserted and deleted lines. Figure 216 shows an example of an UPDPDEL update data set. The first panel shows only the first 80 columns; the second panel shows the result of scrolling right to see the remainder of the member.

```
BROWSE -- FBROWN.UPDATE.PDEL(LINE) ----- LINE 00000000 COL 001 080
COMMAND ==> _                                SCROLL ==> PAGE
***** TOP OF DATA *****
* NEW: FBROWN.TEST2.SCRIPT(DATA)                87/02/24 17.38
* OLD: FBROWN.TEST1.SCRIPT(DATA)
*ID- LINE# SOURCE LINE
      INS#= 00000001
I - 00000001 This line is in the "new" member, but not in the "old".
* RPL#= 00000001 MAT#= 00000001
I - 00000004 This line differs from the text in the "old" member.
D - 00000003 This line differs from the text in the "new" member.
* DEL#= 00000001 MAT#= 00000001
D - 00000005 This line is in the "old" member, but not in the "new".
* END
***** BOTTOM OF DATA *****
```

```
BROWSE -- FBROWN.UPDATE.PDEL(LINE) ----- LINE 00000000 COL 019 098
COMMAND ==> _                                SCROLL ==> PAGE
***** TOP OF DATA *****
TEST2.SCRIPT(DATA)                87/02/24 17.38
TEST1.SCRIPT(DATA)
SOURCE LINE
      INS#= 00000001
line is in the "new" member, but not in the "old".                00010000
      RPL#= 00000001 MAT#= 00000001
line differs from the text in the "old" member.                00040000
line differs from the text in the "new" member.                00030000
      DEL#= 00000001 MAT#= 00000001
line is in the "old" member, but not in the "new".                00050000
***** BOTTOM OF DATA *****
```

Figure 216. UPDPDEL Update Data Set Listing

UPDSEQ0 - Update Sequenced 0

This process option produces a control data set that relates insertions and deletions to the relative line numbers of the “old” data set. It is available for the LINE compare type only. UPDSEQ0 is similar to UPDCMS8, but uses relative line numbers instead of sequence number columns from the “old” data set.

This update data set is characterized by control statements followed by source lines from the “new” data set. All UPDSEQ0 control statements are identified by the characters “./” in columns 1 and 2 of the 80-byte record, followed by one or more blanks and additional blank-delimited fields. The control statements are insert (I), delete (D), replace (R), and comment (*). Control statement data does not extend beyond column 50. Figure 217 shows an example of a UPDSEQ0 update data set listing.

```

BROWSE -- FBROWN.UPDATE.SEQ0(LINE) ----- LINE 00000000 COL 001 080
COMMAND ==> -                               SCROLL ==> PAGE
***** TOP OF DATA *****
./ * NEW: FBROWN.TEST2.SCRIPT(DATA)                87/02/24 17.39
./ * OLD: FBROWN.TEST1.SCRIPT(DATA)
./ R 00000001 00000001 $ 00000002
This line is in the "new" member, but not in the "old".          00010000
This line is reformatted; the spacing in the "new" member differs. 00020000
./ R 00000003 00000003 $ 00000001
This line differs from the text in the "old" member.             00040000
./ D 00000005 00000005 $ 00000000
***** BOTTOM OF DATA *****

```

Figure 217. UPDSEQ0 Update Data Set

UPDSUMO - Update Summary Only

This process option produces an update data set of four lines: “new” data set name, “old” data set name, column headers, and a summary totals line. It is available for the LINE, WORD, and BYTE compare types.

The summary totals line has a “T” in column 1. The summary statistics are located at fixed offsets in the output line. The data set has a line length of 132 bytes.

UPDSUMO Data Set Using LINE Compare

```

BROWSE -- FBROWN.UPDATE.SUMO(LINE) ----- LINE 00000000 COL 001 080
COMMAND ==> -                               SCROLL ==> PAGE
***** TOP OF DATA *****
* NEW: FBROWN.TEST2.SCRIPT                      87/02/24 17.40
* OLD: FBROWN.TEST1.SCRIPT
* NEW-PROC OLD-PROC NEW-INS OLD-DEL TOT-CHG TOT-RFM FI-PROC FI-DIFF N-NOT
T 00000005 00000005 00000002 00000002 00000004 00000001 00000001 00000001 00000
***** BOTTOM OF DATA *****

BROWSE -- FBROWN.UPDATE.SUMO(LINE) ----- LINE 00000000 COL 053 132
COMMAND ==> -                               SCROLL ==> PAGE
***** TOP OF DATA *****
87/02/24 17.40

RFM FI-PROC FI-DIFF N-NOT-PD O-NOT-PD N-DP-LNS O-DP-LNS
0001 00000001 00000001 00000002 00000000 00000000 00000000
***** BOTTOM OF DATA *****

```

Figure 218. UPDSUMO Data Set Using LINE Compare

In Figure 218, the member is shown in split screen mode. The bottom half of the screen shows the result of scrolling right to see the remainder of the member. The following table shows the column numbers used to display the update information:

Column Numbers	Data Item
4-11	Number of new lines processed
13-20	Number of old lines processed
22-29	Number of new line insertions
31-38	Number of old line deletions
40-47	Total number of line changes
49-56	Total number of reformats
58-65	Total number of members processed
67-74	Total number of members different
76-83	Total new members not processed
85-92	Total old members not processed
94-101	Total number of new "do not process" lines
103-110	Total number of old "do not process" lines

Figure 219. UPDSUMO Format Using LINE Compare

UPDSUMO Data Set Using WORD Compare

```

BROWSE -- FBROWN.UPDATE.SUMO(WORD) ----- LINE 00000000 COL 001 080
COMMAND ==> _                                SCROLL ==> PAGE
***** TOP OF DATA *****
* NEW: FBROWN.TEST2.SCRIPT                      87/02/24 17.40
* OLD: FBROWN.TEST1.SCRIPT
* NEW-PROC OLD-PROC NEW-INS OLD-DEL TOT-CHG      FI-PROC FI-DIFF N-NOT
T 00000054 00000053 00000017 00000016 00000030 00000001 00000001 00000
***** BOTTOM OF DATA *****

BROWSE -- FBROWN.UPDATE.SUMO(WORD) ----- LINE 00000000 COL 053 132
COMMAND ==> _                                SCROLL ==> PAGE
***** TOP OF DATA *****
87/02/24 17.40

FI-PROC FI-DIFF N-NOT-PD O-NOT-PD
00000001 00000001 00000002 00000002
***** BOTTOM OF DATA *****

```

Figure 220. UPDSUMO Data Set Using WORD Compare

In Figure 220, the member is shown in split screen mode. The bottom half of the screen shows the result of scrolling right to see the remainder of the member. The following table shows the column numbers used to display the update information:

Column Numbers	Data Item
4-11	Number of new words processed
13-20	Number of old words processed
22-29	Number of new word insertions
31-38	Number of old word deletions
40-47	Total number of word changes
49-56	Blank space. No reformat columns are represented in WORD compare.
58-65	Total number of members processed
67-74	Total number of members different
76-83	Total new members not processed
85-92	Total old members not processed

Figure 221. UPDSUMO Format Using WORD Compare

UPDSUMO Data Set Using BYTE Compare

```

BROWSE -- FBROWN.UPDATE.SUMO(BYTE) ----- LINE 00000000 COL 001 080
COMMAND ==> -                               SCROLL ==> PAGE
***** TOP OF DATA *****
* NEW: FBROWN.TEST2.SCRIPT                      87/02/24 17.41
* OLD: FBROWN.TEST1.SCRIPT
* NEW-PROC OLD-PROC NEW-INS OLD-DEL TOT-CHG      FI-PROC FI-DIFF N-NOT
T 00000400 00000400 00000092 00000092 00000178    00000001 00000001 00000
***** BOTTOM OF DATA *****

BROWSE -- FBROWN.UPDATE.SUMO(BYTE) ----- LINE 00000000 COL 053 132
COMMAND ==> -                               SCROLL ==> PAGE
***** TOP OF DATA *****
87/02/24 17.41

FI-PROC FI-DIFF N-NOT-PD O-NOT-PD
00000001 00000001 00000002 00000002
***** BOTTOM OF DATA *****

```

Figure 222. UPDSUMO Data Set Using BYTE Compare

In Figure 222, the member is shown in split screen mode. The bottom half of the screen shows the result of scrolling right to see the remainder of the member. The following table shows the column numbers used to display the update information:

Column Numbers	Data Item
4-11	Number of new bytes processed
13-20	Number of old bytes processed
22-29	Number of new byte insertions
31-38	Number of old byte deletions
40-47	Total number of byte changes
49-56	Blank space. No reformat columns are represented in BYTE compare.
58-65	Total number of members processed
67-74	Total number of members different
76-83	Total new members not processed
85-92	Total old members not processed

Figure 223. UPDSUMO Format Using BYTE Compare

Process Statements

This section lists the process statements in alphabetic order, showing their syntax, parameter definitions, and examples. All process statements that are shown with a **Compare Format** can be:

- Used in a profile data set with the SuperC utility (option 3.12).
- Entered in a statements data set with the SuperCE utility (option 3.13).

All process statements that are shown with a **Search Format** can be entered in a statements data set and used with the Extended Search-For utility (option 3.13.S), but *not* with the Search-For utility (option 3.14).

Each process statement description contains the following information:

Description	A description of function and operation of the statement. This description may also refer to other statements that can be used with or affected by this statement.
Format	Syntax diagrams for coding the statement, using either Compare Format , Search Format , or both if applicable, according to the conventions listed in "Process Statement Notation Conventions."
Parameters	A description of any required or optional keywords or parameters.
Examples	Sample usage of the statements.

Process Statement Notation Conventions

In this book, the following notation conventions are used to describe the format of the process statements:

- Process statements and keywords are shown in uppercase to help you differentiate between them and their parameters, which are shown in lowercase. However, you can enter them in any case you like.
- Parameters shown in brackets ([]) are optional, with a choice indicated by an OR symbol (|). You can choose one or none; defaults are underscored.
- When specifying a single column for a column range, the colon (:) and the stop-colm parameter to the right of the colon are optional, as indicated by the brackets. For example:

and

SuperC - Process Statements

- Parameters shown without brackets are required. If several parameters are separated by an OR symbol (|), you must select at least one.
- Formats that end with an ellipsis (...) allow you to repeat parameters more than once. However, process statements must be entered completely on one line.
- If a statement contains the KYWD parameter, this keyword must precede any repetition of the parameter(s) that follows it.

CMPBOFS - Compare Byte Offsets

This process statement specifies a starting/ending hexadecimal offset within the input data set for a comparison of a partial data set. It allows programmers to compare portions of one program listing to another and is valid only with a BYTE compare type.

Compare Format

CMPBOFS KYWD hex-offset ...

Parameters

KYWD	Used for data set positioning. You can use one or more of the following:
NTOP	Tells SuperC that the hex-offset value following NTOP is to be considered the top of the “new” data set for comparison purposes. If the comparison is not ended by an NBTM keyword, it continues until the bottom of the data set is reached.
NBTM	Tells SuperC that the hex-offset value following NBTM is to be considered the bottom of the “new” data set for comparison purposes.
OTOP	Tells SuperC that the hex-offset value following OTOP is to be considered the top of the “old” data set for comparison purposes. If the comparison is not ended by an OBTM keyword, it continues until the bottom of the data set is reached.
OBTM	Tells SuperC that the hex-offset value following OBTM is to be considered the bottom of the “old” data set for comparison purposes.
hex-offset	A hexadecimal value from 0 to FFFFFFFF that specifies the top or bottom of a “new” or “old” data set for comparison purposes.

Example

The following statement tells SuperC to compare data between hexadecimal offset 4A0 and hexadecimal offset E2C in both the “old” and “new” data sets.

```
CMPBOFS 4A0 NBTM 4A0 OTOP 4A0 OBTM E2C
```

CMPCOLM - Compare Columns

This process statement compares or searches columns from the input source line(s). Up to six compare and search ranges, and individual columns are allowed. Commas or blank spaces can be used to separate column specifications.

If you enter this statement more than once in the same statements data set, only the last entry is used.

For comparisons, this statement is valid only with the LINE and WORD compare types.

Column numbers specified in this statement are applied to the comparisons of both the "new" and the "old" data sets. Therefore, if you enter the CMPCOLM process statement, you should *not* use either the CMPCOLMN statement or the CMPCOLMO statement.

Compare and Search Format

```
CMPCOLM start-colm[:stop-colm],...
```

Parameters

start-colm A number that identifies the column in which the comparison or search is to begin. All columns to the left of this column are ignored.

stop-colm A number that identifies the column in which the comparison or search is to end for that record. If the data set contains more records, the comparison or search continues with the next record, beginning with the column specified as the start-colm.

This number is separated from the start-colm value by a colon (:). All columns to the right of this column are ignored. If the colon and stop-colm are not entered, only the start-colm value is used. This causes a comparison or search of a single column.

Example

The following statement tells SuperC to compare columns 1-70, 72, and 74.

CMPCOLMN/CMPCOLMO - Compare Column New and Old

These process statements operate the same as CMPCOLM, but refer only to source lines from the “new” input data set (CMPCOLMN) and the “old” data set (CMPCOLMO). They are valid only with the LINE and WORD compare types.

The purpose of these process statements is to let you specify a column(s) to be compared in either the “new” or “old” data set only. Therefore, if you enter the CMPCOLMN process statement, you should *not* use either the CMPCOLM statement or another CMPCOLMN statement in the same statements data set. If you do, SuperC processes both statements, but uses the values in the last CMPCOLM or CMPCOLMN statement only. The same applies to CMPCOLMO.

Also, if you enter a CMPCOLMN statement without a corresponding CMPCOLMO statement, SuperC uses the column(s) specified in the CMPCOLMN statement to compare *both* data sets.

Finally, you should always compare the *same number of columns in both data sets*. For example, suppose the CMPCOLMN statement you enter tells SuperC to compare 15 columns, such as 40 to 55. Here, the corresponding CMPCOLMO statement should also tell SuperC to compare 15 columns, even if they are not all the same, such as 45 to 60. Otherwise, SuperC uses the widest column range specified to compare both data sets.

Compare Format

```
CMPCOLMN start-colm[:stop-colm],...
```

and

```
CMPCOLMO start-colm[:stop-colm],...
```

Parameters

start-colm A number that identifies the column in which the comparison is to begin. All columns to the left of this column are ignored.

stop-colm A number that identifies the column in which the comparison is to end for that record. If the data set contains more records, the comparison continues with the next record, beginning with the column specified as the start-colm.

This number is separated from the start-colm value by a colon (:). All columns to the right of this column are ignored. If the colon and stop-colm are not entered, only the start-colm value is used. This causes a comparison of a single column.

Example

The following statements:

tell SuperC to compare the data in columns 1-70, 72, and 74 in the “new” data set to columns 4-73, 75, and 77 in the “old” data set.

CMPLINE - Compare Line

This process statement specifies a starting and ending line within the input data set for a compare or search of a partial data set. The starting and ending points can be specified as a line number or as a character string. The statement can be specified on one complete line or can have separate CMPLINE statements for each keyword used.

For comparisons, this statement is valid only with the LINE and WORD compare types.

Compare and Search Format

```
CMPLINE KYWD line-number|'string'[,start-coll[:stop-coll]],...
```

Parameters

KYWD	Used for data set positioning. You can use one or more of the following: <i>Note: Only NTOP and NBTM are valid for searches.</i>
NTOP	Shows that the following values are to identify the top of a data set that is to be compared or searched. For comparisons, this means the top of the “new” data set. If the comparison or search is not ended by an NBTM keyword, it continues until the bottom of the data set is reached.
NBTM	Shows that the following values are to identify the bottom of a data set that is to be compared or searched. For comparisons, this means the bottom of the “new” data set.
OTOP	Shows that the following values are to identify the top of the “old” data set for comparison purposes. If the comparison is not ended by an OBTM keyword, it continues until the bottom of the data set is reached.
OBTM	Shows that the following values are to identify the bottom of the “old” data set for comparison purposes.
line-number	The number of a line in the data set(s) to be compared or searched. This line is considered to be either the top or bottom of the data set, depending on whether it follows NTOP, NBTM, OTOP, or OBTM, respectively.
'string'	A character string. Single quotation marks are required. If the character string contains a single quotation mark, you must enter it in the process statement as two single quotation marks.

The line in which this string exists is considered to be either the top or bottom of a data set, depending on whether it follows NTOP, NBTM, OTOP, or OBTM, respectively.

start-colm A number that identifies the column in which SuperC is to begin its search for 'string'. All columns to the left of this column are ignored.

stop-colm A number that identifies the column in which SuperC is to end its search for 'string', but only for that record. If the data set contains more records, the search continues with the next record, beginning with the column specified as the start-colm.

This number is separated from the start-colm value by a colon (:). All columns to the right of this column are ignored. If the colon and stop-colm are not entered, only the start-colm value is used. This causes a search of a single column for the first character of 'string'.

All columns to the right of the column that the last character of 'string' can occupy are ignored. For example, if you specify 'ABCD' as a string that must exist between columns 5 and 69, the search really includes columns 5 to 72. Therefore, the string can be found even if the first character is in column 69, with the others in columns 70, 71, and 72.

Examples

Here are some examples, with explanations in the comment lines to the right.

```

CMPLINE TOP 55,99,100 /* Identifies line 55 as the */
                        /* top and line 99 as the */
                        /* bottom of the "new" data */
                        /* set */
CMPLINE TOP 1,100 /* Identifies the first line */
                  /* in the "old" data set */
                  /* that contains 'ABCD' as */
                  /* the top */
CMPLINE TOP 1,100,5:72 /* Identifies the first line */
                      /* in the "old" data set */
                      /* that contains 'ABCD' */
                      /* between columns 5 and 72 */
                      /* as the top */

```


Comment Lines

You can enter comment lines in your statements data set. A comment line can begin with either:

- An asterisk and a blank as the first two characters on a process statement line
- A period, an asterisk, and a blank as the first three characters on a process statement line.

The difference between the two types of comments is that “.*” is never printed in the listing, while the “*” comment is always printed unless the **SLIST off** process statement is used.

Compare and Search Format

* Text of comment

and

.* Text of comment

DPLINE/DPLINEC - Don't Process This Line

These process statements exclude from the comparison or search any line that contains the character string you specify. You can also specify the column in which the character string begins or a range of columns within which the character string may be found.

DPLINEC is a continuation of the DPLINE that immediately precedes it and represents a logical "AND." More than one DPLINE and DPLINEC statement can be used.

For comparisons, these statements are valid only with the LINE and WORD compare types.

Compare and Search Format

```
DPLINE 'string'[,start-colm[:stop-colm]]
```

and

```
DPLINEC 'string'[,start-colm[:stop-colm]]
```

Parameters

'string'	<p>The character string that exists on a line that you want to be excluded from the comparison or search. Single quotation marks are required. If the character string contains a single quotation mark, you must enter it in the process statement as two single quotation marks.</p> <p>Only the string is required. If you do not enter a column(s), any line that contains the character string is excluded.</p>
start-colm	<p>A number that identifies the column in which SuperC is to begin looking for 'string'. All columns to the left of this column are ignored.</p>
stop-colm	<p>A number that identifies the column in which SuperC is to end its search for 'string', but only for that record. If the data set contains more records, the search continues with the next record, beginning with the column specified as the start-colm.</p> <p>This number is separated from the start-colm value by a colon (:). All columns to the right of this column are ignored. If the colon and stop-colm are not entered, only the start-colm value is used. This causes a search of a single column for the first character of 'string'.</p> <p>All columns to the right of the column that the last character of 'string' can occupy are ignored. For example, if you specify 'ABCD' as a string that must exist between columns 5 and 69, the search really includes columns 5 to 72. Therefore, the string can be found even if the first</p>

character is in column 69, with the others in columns 70, 71, and 72.

Examples

Here are some examples, with explanations in comment lines to the right.

```

/* Prevents the processing */
/* of lines in which 'ABCD' */
/* is found */

/* Continues the preceding */
/* DPLINE statement by */
/* preventing the processing */
/* of lines that contain */
/* both 'ABCD' and 'EFGH' */

/* Prevents the processing */
/* of lines in which the */
/* character equivalent of */
/* X'C1C27BF1' (AB,1) is */
/* found, but only if it is */
/* found between columns 5 */
/* and 72 */

```

LNCT

LNCT - Line Count

This process statement lets you specify the maximum number of lines per page for your output listing data set. For comparisons, it is valid with all compare types.

Compare and Search Format

LNCT nnnnnn

Parameters

nnnnnn The number of lines per page. A page heading is generated for each group of lines.

If you are printing a listing, the maximum setting you can specify depends on your printer and the size of the paper you are using. However, for seeing the listing in Browse mode, the setting can be higher.

Any number outside this setting results in a warning at the bottom of the output listing. Also, the LNCT statement you entered is ignored and the default line count of 55 lines per page is used.

Example

The following statement limits each page of the output listing to 60 lines:

LSTCOLM - List Columns

This process statement selects a column(s) from the source data to be listed in the output. Only a single start-stop range is allowed.

All source data between these two columns is included in the listing data set, but only if the column selections are no wider than the output listing line allocated. Excess columns are ignored. The widths can be:

- For comparisons, NARROW (55), WIDE (80), normal (106), or LONGLN (176).
- For searches, normal (106) or LONGLN (176).

For comparisons, this statement is valid for the LINE compare type only.

Compare and Search Format

```
LSTCOLM start-colm[:stop-colm]
```

Parameters

- | | |
|-------------------|---|
| start-colm | A number that identifies the column in which SuperC is to begin searching the source data set for data to include in the listing. All columns to the left of this column are ignored. |
| stop-colm | A number that identifies the column in which the listing of source data is to end for that record. If the data set contains more records, the listing of source data continues with the next record, beginning with the column specified as the start-colm. |

This number is separated from the start-colm value by a colon (:). All columns to the right of this column are ignored. If the colon and stop-colm are not entered, only the start-colm value is used. This causes the listing of only a single column of the source data set.

Example

The following statement lists source data in columns 2-82:

```
LSTCOLM 2:82
```

NCHGT/OCHGT - New and Old Change Text

These process statements change text in a data set from 'find-mask' to 'string' before a comparison or search begins. You can also specify a column range to search or even a single column. More than one statement can be used.

The 'find-mask' and 'string' parameters do not have to be the same length. In fact, the 'string' parameter can consist of blanks. Both quote ('xxx') and hex (x'01ab') notation are allowed. Double quotation marks in either a 'find-mask' or 'string' are interpreted as single quotation marks.

For comparisons, NCHGT changes text in the "new" data set and OCHGT changes text in the "old" data set. Also, these statements are valid only with the LINE and WORD compare types.

For searches, only the NCHGT statement is valid.

Compare Format

```
NCHGT 'find-mask','string'[ ,start-colm[:stop-colm]]
```

and

```
OCHGT 'find-mask','string'[ ,start-colm[:stop-colm]]
```

Search Format

```
NCHGT 'find-mask','string'[ ,start-colm[:stop-colm]]
```

Parameters

'find-mask' A character string that is to be replaced by the contents of 'string' before a comparison or search. Single quotation marks are required. If the 'find-mask' contains a single quotation mark, you must enter it in the process statement as two single quotation marks.

A question mark(s) can be included in the 'find-mask'. It shows that the position it occupies can be filled by any character, one character per question mark.

The 'find-mask' is not considered to have blank spaces as delimiters. Therefore, to avoid replacing a string that may be found within another string, insert blanks within the single quotation marks.

'string' A character string that is to replace the contents of 'find-mask' before a comparison or search. Single quotation marks are required. If the character string contains a single quotation mark, you must enter it in the process statement as two single quotation marks.

A question mark(s) can be included in the 'string'. It shows that the position it occupies can be filled by any character, one character per question mark.

If there is not enough space for the string to be substituted, the substitution does not occur. Instead, a warning message is printed at the bottom of the listing.

start-colm A number that identifies the column in which SuperC is to begin looking for 'find-mask'. All columns to the left of this column are ignored.

stop-colm A number that identifies the column in which SuperC is to end its search for 'find-mask', but only for that record. If the data set contains more records, the search continues with the next record, beginning with the column specified as the start-colm.

This number is separated from the start-colm value by a colon (:). All columns to the right of this column are ignored. If the colon and stop-colm are not entered, only the start-colm value is used. This causes a search of a single column for the first character of 'find-mask'.

All columns to the right of the column that the last character of 'find-mask' can occupy are ignored. For example, if you specify 'ABCD' as a 'find-mask' that must exist between columns 5 and 69, the search really includes columns 5 to 72. Therefore, 'find-mask' can be found even if the first character is in column 69, with the others in columns 70, 71, and 72.

Examples

Here are some examples, with explanations in comment lines to the right. Examples of the NCHGT statement apply to both comparisons and searches, but OCHGT examples apply to comparisons only.

```
NCHGT 'ABCD','XXXX'          /* In the "new" data set,    */
                             /* replaces each 'ABCD' with */
                             /* 'XXXX'                    */

NCHGT 'ABCD','',1:50          /* In the "new" data set,    */
                             /* replaces each 'ABCD' with */
                             /* a blank space, but only   */
                             /* between columns 1 and 53  */

OCHGT 'EF??','XX'            /* In the "old" data set,    */
                             /* replaces each occurrence  */
                             /* of 'EF' followed by any   */
                             /* two characters with 'XX'  */

NCHGT 'X'7B01','':1',1:50    /* In the "new" data set,    */
                             /* replaces each occurrence  */
                             /* of hexadecimal string     */
                             /* X'7B01' with '':1', but    */
                             /* only between columns 1     */
                             /* and 51                      */
```

NTITLE/OTITLE - Alternate Title for Input Data Sets

These process statements let you specify an alternate data set title for data sets being compared or searched. The name appears in the output list data set.

OTITLE is valid for comparisons only. Both statements are valid for all compare types.

Compare Format

```
NTITLE 'new-title-text'
```

and

```
OTITLE 'new-title-text'
```

Search Format

```
NTITLE 'new-title-text'
```

Parameter

'new-title-text'	The alternate name for the data set. You can use any name, but it must be enclosed in single quotation marks. No check is made to see if the name follows TSO naming conventions.
-------------------------	---

Example

The following statement give the “new” or search data set the alternate name 'PDFPROJ.ALTNAME.SCRIPT'.

```
NTITLE 'PDFPROJ.ALTNAME.SCRIPT'
```


SELECT - Select PDS Members

This process statement lets you specify members to select from a PDS. You can specify as many member names as will fit on one line. Therefore, if you need to select additional members, enter a new SELECT statement.

For comparisons, the “new” members are normally compared with “old” members that have the same names. Use the colon character (:) to compare members that are not named alike. SELECT is valid with all compare types.

Compare Format

```
SELECT new-member[:old-member],...
```

Search Format

```
SELECT member,...
```

Compare Parameters

new-member The name of a “new” PDS member that is to be compared to an “old” PDS member.

old-member The name of an “old” PDS member that does not have a like-named member in the “new” PDS. This member name, if entered, must be separated from the new-member name by a colon (:).

If the old-member name is not used, SuperC attempts to compare the new-member to a like-named member of the “old” PDS.

Search Parameter

member The name of a PDS member that you want to search for a character string.

Examples

The following show the difference between using the SELECT statement for comparisons and using it for searches.

Compare Example:

The following statement:

```
SELECT PROG1,NEWPROG2:OLDPROG2,PROG3
```

tells SuperC to make the following one-to-one comparisons:

NEW MEMBER		OLD MEMBER
-----		-----
PROG1	----->	PROG1
NEWPROG2	----->	OLDPROG2
PROG3	----->	PROG3

SELECT

Search Example:

The following statement:

selects PDS members PROG1, NEWPROG2, OLDPROG2, and PROG3 to be searched.

SLIST - Include Statements in Listing

This process statement lets you decide whether to include process statements in an output listing.

Compare and Search Format

SLIST on | off

Parameters

- | | |
|------------------|---|
| <u>on</u> | Includes process statements in the output listing. The <u>on</u> parameter is the default, so the listing includes all the process statements that are used unless you specify SLIST off . |
| off | Prevents process statements from being included in the output listing. |

Example

SLIST affects only the process statements that follow it. For example, if you do not want to include any process statements in a listing, enter:

```
SLIST off
```

as the first process statement in your statements data set.

SRCHFOR/SRCHFORC - Search-For String

These process statements let you specify a string that you want to find. The 'string' parameter is required. You can also include a search type to qualify the search. More than one SRCHFOR and SRCHFORC statement can be used in a statements data set.

SRCHFORC is a continuation of the SRCHFOR or SRCHFORC statement that immediately precedes it and represents a logical "AND." That is, all the strings within a SRCHFOR/SRCHFORC group must exist on the same line. Both SRCHFOR and SRCHFORC override any character strings that you type in a CAPS or ASIS field on the Extended Search-For Utility panel.

Search Format

```
SRCHFOR  'string'[,qualifier]

and

SRCHFORC 'string'[,qualifier]
```

Parameters

'string'	The character string to be searched for. Quotation marks are required before and after the string. If the character string contains a single quotation mark, you must enter it in the process statement as two single quotation marks.
qualifier	One of the search types listed below. See "Search-For Strings and Keywords" on page 201 for definitions of these search types.
P	Prefix.
W	Word.
S	Suffix.
blank	Any occurrence of the string.

Examples

Here are some examples, with explanations in comment lines to the right.

```
SRCHFOR 'ABC' /* Tells SuperC to find all */
               /* occurrences of 'ABC' */

SRCHFORC 'DEF' /* Continues the preceding */
               /* SRCHFOR statement by */
               /* searching for lines that */
               /* contain both 'ABC' and */
               /* 'DEF' */
```

Index

Special Characters

.IM control word 211
 <FIELD> statement 164
 (ACCOUNT) parameter
 JOB STATEMENT INFORMATION 56
 | (OR symbol) 15
 & (command prefix)
 displaying commands after execution 87
 &xxxxx action
 ACTION column 161
 * (comment line) process statement
 Extended Search-For utility (option 3.13.S) 401
 SuperCE utility (option 3.13) 401
 * (in source listing) 340
 QUAL field 320
)ATTR section 164
)BODY section 164
)INIT section 164
 / (data set name character)
 Data Set List utility (option 3.4) 132
 /* lines
 JOB STATEMENT INFORMATION 56
 //OUTDD DD field 178
 % field 130
 %USED field 137
 %USED keyword
 SORT command 146
 ? (invalid load module directory fields) 42
 ? (more than 16 extents)
 FREE EXTENTS field 137
 'NAME' parameter
 JOB STATEMENT INFORMATION 56
 = (jump function) 13, 25
 = (repeat last command) line command
 Data Set List utility (option 3.4) 141
 "Target" Panel for Promoting 121
 "To" Panel for Copying 116
 "To" Panel for Moving 118
 [] (optional operands) 15
 { } (one operand required) 15

A

A (ANSI) parameter
 PRINTER CARRIAGE CONTROL field 156
 A (ASA printer characters) record format
 A (variable attributes) field 294, 300, 302, 304

AC field 42
 ACCOUNT command
 TSO 32, 108
 ACCOUNT parameter
 JOB STATEMENT INFORMATION 56
 ACTION column 160
 ACTION field 121
 Activate/Create Profile DS Name field 197
 activating library controls 34
 ACTIVE field 316, 318, 320
 Add Row panel 304
 Additional IBM Program Development Products
 panel 329
 ADDITIONAL INPUT LIBRARIES field 216
 AFTER field 324
 AFTER parameter
 Breakpoint Primary Option Menu 324
 WHEN field 320
 ALARM indicator 165
 ALIAS action
 ACTION column 161
 alias as substitute for PDF 4, 5
 ALIAS-OF field 42
 ALL parameter
 FIND command 91
 FUNCTION field 316
 VARIABLE field 317
 ALLOCATE command
 TSO 107
 allocating libraries 29, 33
 American National Standard Institute 156
 ANSI
 See American National Standard Institute
 ANYC process option
 Extended Search-For utility (option 3.13.S) 379
 SuperCE utility (option 3.13) 379
 APNDLST process option
 Extended Search-For utility (option 3.13.S) 381
 SuperCE utility (option 3.13) 381
 APNDUPD process option
 SuperCE utility (option 3.13) 383
 APPLICATION field 324
 APPLICATION ID field 159
 ASIS parameter
 COMMAND LINE PLACEMENT field 74
 Assembler 216
 Assembler H (option 4.1) 219
 allocation data sets
 SYSIN 367
 SYSLIB 367
 SYSLIN 368
 SYSPRINT 368
 SYSPUNCH 369

SYSTEM 368
 SYSUT1 369
 Assembler H (option 5.1) 258
 ASSEMBLER OPTIONS field
 Foreground Assembler H (option 4.1) 219
 Foreground Assembler XF (option 4.1A) 220
 Assembler XF (option 4.1A) 220
 Assembler XF (option 5.1A) 260
 asterisk (*) parameter
 SELECT command 45
 ATTENTION (PA1) key 18
 ATTRIB display view
 INITIAL DISPLAY VIEW field 126
 ATTRIBUTES field 42

B

B (batch) parameter
 MODE field 173
 B (blocked) record format
 RECORD FORMAT field 109
 B (browse data set) line command
 Data Set List utility (option 3.4) 138
 B (browse) line command
 Data Set List utility (option 3.4) 47
 Library utility (option 3.1) 47
 Batch (option 5) 251
 Assembler H (option 5.1) 258
 Assembler XF (option 5.1A) 260
 JCL generation for Assembler and Linkage
 Editor 257
 JCL generation for compilers 256
 Linkage Editor (option 5.7) 270
 Member Parts List (option 5.12) 273
 OS/VS COBOL compiler (option 5.2A) 264
 output listings 54
 overview 6
 Pascal/VS compiler (option 5.6) 269
 password protection, lack of 253
 PL/I checkout compiler (option 5.4) 267
 PL/I optimizing compiler (option 5.5) 268
 processing sequence 251
 VS COBOL II compiler (option 5.2) 262
 VS COBOL II interactive debug (option
 5.10) 272
 VS FORTRAN compiler (option 5.3) 265
 Batch Assembler H panel 258
 Batch Assembler XF panel 260
 Batch Linkage Edit panel 270
 Batch Member Parts List panel 273
 Batch OS/VS COBOL Compile panel 264
 Batch Pascal/VS Compile panel 269
 Batch PL/I Checkout Compile panel 267
 Batch PL/I Optimizing Compile panel 268
 Batch Selection Panel 251
 Batch Selection Panel With JCL Generated 255
 Batch VS COBOL II Compile panel 262
 Batch VS FORTRAN Compile panel 265

BEFORE field 324
 BEFORE parameter
 Breakpoint Primary Option Menu 324
 WHEN field 320
 Blank action
 ACTION column 161
 blank options 9
 blank parameter
 ACTIVE field 316, 318, 320
 DSORG field 131
 MULTIPLE STRINGS field 200
 OPERATION field 318
 PACK DATA field 122
 POOL field 318
 PRINTER CARRIAGE CONTROL field 156
 SEQUENCE NUMBERS field 176
 SPECIFY PACK OPTION FOR "TO" DATA SET
 field 117
 SRCHFOR process statement 412
 WHEN field 320
 BLKS parameter
 SPACE UNITS field 108
 BLKSZ field 131
 BLKSZ keyword
 SORT command 146
 BLOCK SIZE field 110
 BOTTOM parameter
 BY ROW NUMBER field 298
 COMMAND LINE PLACEMENT field 74
 BREAKPOINT field 324
 Breakpoint Primary Option Menu 323
 Breakpoints (option 7.8)
 control display 322
 defined 319
 finding a breakpoint
 Cancel (option C) 323
 Go (option G) 323
 input errors 322
 line commands
 D (delete) 320
 I (insert) 320
 R (repeat) 320
 primary commands
 CANCEL 320
 END 320, 322
 LOCATE 320
 QUAL 320
 qualification 322
 qualification parameter values 320
 specifying breakpoints 319
 syntax checking 322
 Breakpoints panel 319
 with qualification parameter values 321
 Browse - Entry Panel 85
 Browse (option 1) 85
 Browse commands
 BROWSE 87
 COLUMNS 88
 DISPLAY 89
 displaying after execution 87

- FIND 90
- HEX 95
- LOCATE 96
- RESET 97
- Browse data display 86
- browsing DBCS data as EBCDIC data 51
- current position in data set 96
- ending Browse 87
- mixed mode 51
- overview 6
- saving scroll value 27
- selecting Browse 85
- valid logical record length 85
- valid record format 85
- BROWSE command
 - Browse (option 1) 87
 - parameter
 - member 88
- Browse Command Panel 88
- Browse data display 86
- BROWSE field 234
- Browse hexadecimal display
 - data 96
 - vertical 96
- BROWSE Output field 186
- BY ROW NUMBER field 298
- BY VARIABLE field 299
- bypassing menus 5
- BYTE parameter
 - Compare Type field 184

C

- C (catalog data set) line command
 - Data Set List utility (option 3.4) 139
- C (continuation) keyword
 - Search-For utility (option 3.14) 202
- CALL statement format 245
- Canadian French terminals 65
- Cancel (option C) 323
- CANCEL command
 - Breakpoints (option 7.8) 320
 - Command Table utility (option 3.9) 162
 - Dialog Test (option 7) 284
 - Tables (option 7.4) 303, 305
 - Traces (option 7.7) 317, 318
 - Variables (option 7.3) 294
- cataloged partitioned data set 33
- CC parameter
 - DISPLAY command 89
- CHANGED field 40
- Changes (option C)
 - overview 7
- char parameter
 - DISPLAY command 89
- character strings, use of 93
- character translation for APL and TEXT
 - keyboards 803

- CHARS parameter
 - FIND command 90
- CHG operation
 - OPERATION field 318
- CHNG parameter
 - LISTING TYPE field 175, 184
- CLASS field 156
- CLEAR key 19
- CLIST
 - attention exits 277
 - error exits 277
- CMD field 288
- CMPBOFS process statement
 - keywords
 - NBTM 395, 399
 - NTOP 395, 399
 - OBTM 395, 399
 - OTOP 395, 399
 - parameters
 - hex-offset 395
 - KYWD 395
 - SuperCE utility (option 3.13) 395
- CMPCOLM process statement
 - Extended Search-For utility (option 3.13.S) 396
 - parameters
 - start-colm 396, 397
 - stop-colm 396, 397
 - SuperCE utility (option 3.13) 396
- CMPCOLMN process statement
 - SuperCE utility (option 3.13) 397
- CMPCOLMO process statement
 - SuperCE utility (option 3.13) 397
- CMPLINE process statement
 - Extended Search-For utility (option 3.13.S) 399
 - parameters
 - 'string' 399
 - KYWD 399
 - start-colm 400
 - stop-colm 400
 - SuperCE utility (option 3.13) 399
- CNPML process option
 - SuperCE utility (option 3.13) 381
- COBOL interactive debug (option 4.10A) 241
 - print output data sets 243
 - processing sequence 241
 - symbolic debug data sets 242
- COBOL Interactive Debug panel 241
- COBOL parameter
 - SEQUENCE NUMBERS field 176
- COBOL process option
 - Extended Search-For utility (option 3.13.S) 379
 - SuperCE utility (option 3.13) 379
- COBOL Structuring Facility panel 333, 334
- COBOL/SF-B (option 9.5)
 - COBOL Structuring Facility background
 - dialog 334
- COBOL/SF-F (option 9.4)
 - COBOL Structuring Facility foreground
 - dialog 333
- code, defined 9
- col-1 and col-2 parameters

- FIND command 90
- column-identification line 88
 - removing with RESET command 97
- column limitations 94
- COLUMNS command
 - Browse (option 1) 88
- Command (option 6) 275
 - interrupting a TSO command or CLIST 277
 - overview 6
 - restrictions for entering a CLIST 277
 - Session Manager, using 277
 - terminal settings 278
 - TSO commands and CLISTs, entering 276
- COMMAND DELIMITER field 66
- command error processing 16
- COMMAND field 8, 14, 129
- COMMAND LINE PLACEMENT field 74
- command name abbreviations 365
- command prefix (&)
 - displaying commands after execution 87
- Command Table Editing Panel 160
- Command Table utility (option 3.9)
 - line commands
 - D (delete) 162
 - I (insert) 162
 - R (repeat) 162
 - primary commands
 - CANCEL 162
 - END 162
- Command Table Utility panel 159
- commands
 - default PF key settings 19
 - displaying after execution 87
 - entering 8, 14
 - entering long commands 275
 - entering with PF keys 19
 - error processing 16
 - ISPF 12
 - ISPF/PDF notation 15
 - line 6, 14
 - PDF command 4
 - using an alias 4, 5
 - using option parameter 5
 - primary 6, 14
 - retrieving 27
 - scroll 26
 - stacking with delimiter 16
 - TSO 22
- commands, Dialog Test 282
- commands, entering 14
 - command error processing 16
 - COMMAND field 14
 - ISPF/PDF command notation 15
 - PF keys 19
- Compare Type field 183
- COMPILER OPTIONS field
 - Batch Pascal/VS compiler (option 5.6) 269
 - Batch PL/I optimizing compiler (option 5.5) 268
- compress request exit 103, 141
- concatenation
 - during editing 35
 - during language processing 35
- COND parameter
 - BROWSE Output field 186
- CONFIRM command
 - Data Set List utility (option 3.4) 142
 - parameters
 - OFF 142
 - ON 142
- Confirm Delete panel 112
- CONFIRM DELETE REQUEST field 127
- Confirm Purge panel 113
- Confirm Table Row Delete panel 301
- control file 327
- controlling ISPF libraries with LMF 32
 - library concatenation 34
 - library member change and promotion 36
- Convert Menus/Messages panel 163
 - copy panel 164
- Convert Menus/Messages utility (option 3.10)
 - options
 - 1 - convert old format menus to new format panels 164
 - 2 - convert old format messages to new format messages 165
- COPY command
 - TSO 159
- Copy Format Selection List panel 169
- COPY statement 211
- copying and locking members 119
- CREATED field 40, 131
- CREATED keyword
 - SORT command 146
- Cross System Product/Application Development panel 330
- Cross System Product/Application Execution panel 331
- CSP/AD (option 9.1)
 - Cross System Product/Application Development 330
- CSP/AE (option 9.2)
 - Cross System Product/Application Execution 331
- CURRENT ROW COUNT field 308, 309
- CURRENT ROW field 298
- CURRENT ROW POINTER field 306, 309
- CURRENT STATUS field 324
- CURSOR command
 - ISPF 13
- CURSOR FIELD field 291
- cursor movement keys 22
- CURSOR parameter
 - SCROLL field 8
- CURSOR POSITION field 291
- CYLS parameter
 - SPACE UNITS field 108

D

D (DBCS) parameter
 FIELD TYPE field 167

D (delete) line command
 Breakpoints (option 7.8) 320
 Command Table utility (option 3.9) 162
 Data Set List utility (option 3.4) 47, 138
 Dialog Test (option 7) 285
 Format Specification utility (option 3.11) 170
 Library utility (option 3.1) 47
 SCRIPT/VS (option 4.9) 237
 Tables (option 7.4) 303, 305
 Traces (option 7.7) 317, 318
 Variables (option 7.3) 294

D (delete) parameter
 Process option field 24, 67

DA data set organization
 DSORG field 131

DASD
 See direct access storage device

data entry panels 9

DATA parameter
 CMLINE process statement 399
 HEX command 95
 SCROLL field 8

data set allocation exit 107

data set control blocks 137

data set format 10

Data Set List utility (option 3.4)
 data set list exit 124
 data set list listings 347

display views and panels
 ATTRIB 126, 129
 sequence of display views 126
 SPACE 126, 128
 TOTAL 126, 129
 VOLUME 126, 128

line commands
 = (repeat last command) 141
 B (browse data set) 138
 C (catalog data set) 139
 D (delete data set) 138
 E (edit data set) 138
 F (free unused space) 141
 I (data set information) 139
 M (display member list) 139
 P (print data set) 139
 R (rename data set) 139
 S (information (short)) 139
 U (uncatalog data set) 139
 X (print index listing) 139
 Z (compress data set) 141

list data set 143

member list line commands 47

options
 Blank - display data set list 127
 P - print data set list 135

V - display VTOC information 136

primary commands
 CONFIRM 142
 FIND 142
 LOCATE 143
 RFIND 142
 SAVE 143
 SHOWCMD 144
 SORT 146

saving scroll value 27

source and index listings 339

Data Set List Utility panel 124

Data Set List Utility SHOWCMD panel 145

DATA SET NAME field 31, 178, 246

data set organization 131

DATA SET PASSWORD field 49

Data Set utility (option 3.2)
 options
 A - allocate new data set 106
 Blank - data set information 114
 C - catalog data set 111
 D - delete entire data set 112
 R - rename entire data set 111
 S - data set information (short) 114
 U - uncatalog data set 112

DATE CREATED field 307, 309

DAU data set organization
 DSORG field 131

DBCS
 See double-byte character set

DBCS COLUMN SPECIFICATION field 299

DCF
 See Document Composition Facility

DDDD format
 EXPIRATION DATE field 111

DEBUG COMMAND DATA SET field
 Batch VS COBOL II interactive debug (option 5.10) 272

DEBUG OPTIONS field
 Foreground FORTRAN interactive debug (option 4.11) 244

default mode 7

default PF key settings 20
 ending PDF or a PDF function 22
 getting help 20
 repeating a FIND or CHANGE command 25
 scrolling within a data display 25
 splitting and swapping display screens 21

DELETE command
 TSO 127, 128, 142

delimiters
 used to stack commands 16

DELTA parameter
 LISTING TYPE field 175, 184

DESCRIPTION column 161

Description field 238

DEVICE field 131

DEVICE keyword
 SORT command 146

Dialog Services (option 7.6)
 invoking a dialog service 313

DIALOG SERVICES TO BE TRACED field 316
 Dialog Test (option 7) 279
 Breakpoints (option 7.8) 319
 commands 282
 Dialog Services (option 7.6) 313
 effect of RETURN command 24
 environment 280
 accessing and updating variables 280
 available ISPF facilities 280
 exceptions to restoration when using Go (option G) 324
 ISPF log generation 280
 error handling, severe 282
 Exit (option 7.X) 326
 exiting with jump function (=) 25
 Functions (option 7.1) 287
 line commands
 D (delete lines) 285
 I (insert lines) 285
 R (repeat lines) 286
 Log (option 7.5) 310
 overview 6
 Panels (option 7.2) 291
 primary commands
 CANCEL 284
 END 284
 LOCATE 284
 QUAL 285
 RESUME 285
 severe error handling
 message displayed 282
 reasons errors occur 282
 Tables (option 7.4) 298
 Traces (option 7.7) 315
 Tutorial (option 7.T) 326
 usual test methods 279
 variable usage 281
 Variables (option 7.3) 293
 why you should generate a log data set 69
 Dialog Test Primary Option Menu 280
 Dialog Test Tutorial panel 326
 dialog, definition of 1
 direct access storage device
 using with packed data 51
 DIRECTORY BLOCKS field 109
 Display (option 0.4)
 specifying ASIS 74
 specifying BOTTOM 74
 Display Characteristics panel 74
 DISPLAY command
 Browse (option 1) 89
 parameters
 CC 89
 char 89
 NOCC 89
 display format 7
 Display Panel 291
 Display Row panel 300
 DISPLAY STYLE OPTIONS field 234
 DLMDUP process option

DLREFM process option
 SuperCE utility (option 3.13) 381
 Document Composition Facility 233
 double-byte character set
 Browse or Edit DBCS data as EBCDIC data 51
 column specification in dialog test 299
 defining data type 167
 Dialog Test variables format 297
 Edit and Browse 51
 effect on SuperC utilities 374
 finding a one-byte hexadecimal string 94
 format definition 50
 Format Specification utility (option 3.11) 50, 166
 formatted data Edit and Browse 50
 hexadecimal format 51
 invalid mixed data 51
 MIXED MODE field 50
 using the VALUE field 299
 DOWN command
 ISPF 12, 26
 DPACMT process option
 Extended Search-For utility (option 3.13.S) 380
 SuperCE utility (option 3.13) 380
 DPADCMT process option
 Extended Search-For utility (option 3.13.S) 380
 SuperCE utility (option 3.13) 380
 DPBLKCL process option
 Extended Search-For utility (option 3.13.S) 380
 SuperCE utility (option 3.13) 380
 DPCBCMT process option
 Extended Search-For utility (option 3.13.S) 380
 SuperCE utility (option 3.13) 380
 DPFTCMT process option
 Extended Search-For utility (option 3.13.S) 380
 SuperCE utility (option 3.13) 380
 DPLINE process statement
 Extended Search-For utility (option 3.13.S) 402
 parameters
 'string' 402
 start-colm 402, 407
 stop-colm 402, 407
 SuperCE utility (option 3.13) 402
 DPLINEC process statement
 Extended Search-For utility (option 3.13.S) 402
 SuperCE utility (option 3.13) 402
 DPPLCMT process option
 Extended Search-For utility (option 3.13.S) 380
 SuperCE utility (option 3.13) 380
 DPPSCMT process option
 Extended Search-For utility (option 3.13.S) 380
 SuperCE utility (option 3.13) 380
 DSCBs
 See data set control blocks
 DSLIST exit 124
 DSNNAME LEVEL field 124
 DSORG field 131
 DSORG keyword
 SORT command 146
 DS1-DS4 fields 186, 191

E

E (EBCDIC) parameter
 FIELD TYPE field 167
 E (edit data set) line command
 Data Set List utility (option 3.4) 138
 EBCDIC
 See extended binary coded decimal interchange code
 Edit - Entry Panel 99
 Edit (option 2) 99
 editing a data set 99
 editing DBCS data as EBCDIC data 51
 mixed mode 51
 overview 6
 saving scroll value 27
 valid logical record length 99
 valid record format 99
 EDIT command
 Data Set List utility (option 3.4) 47
 END command
 Breakpoints (option 7.8) 320, 322
 Command Table utility (option 3.9) 162
 Dialog Test (option 7) 284
 ISPF 12, 23
 Tables (option 7.4) 301, 303, 305
 Traces (option 7.7) 317, 318
 Variables (option 7.3) 294
 ending PDF or a PDF function
 END command 23
 Exit option (X) 25
 RETURN command 24
 ENTER SESSION MANAGER MODE field 277
 ENTER TSO COMMAND OR CLIST BELOW
 field 275
 entering commands with PA and PF keys
 program access (PA) keys 18
 program function (PF) keys 19
 ENVIRON command
 ISPF 13, 81
 environment, Dialog Test 280
 ERASE INPUT key 19
 error handling, Dialog Test 282
 errors
 trigger statement 215
 Example of Using Aspect Ratio Parameter 0 79
 Example of Using Aspect Ratio Parameter 1 80
 examples
 = (repeat last command) line command 141
 account information 56
 additional input libraries 216
 aspect ratio parameter 0 79
 aspect ratio parameter 1 80
 Assembler H allocation data sets
 SYSIN 367
 SYSLIB 367
 SYSLIN 368
 SYSPPRINT 368
 SYSPPRINT 368

SYSTEM 368
 SYSUT1 369
 BROWSE command 88
 bypassing menus 5
 CMPBOFS process statement 395
 CMPCOLM process statement 396
 CMPCOLMN process statement 398
 CMPCOLMO process statement 398
 CMPLINE process statement 400
 column-identification line 89
 command error processing 16
 command interpretation 276
 command table usage restriction 159
 CONFIRM command 142
 Copy Format Selection List panel 169
 data entry panel 85
 data set information displays 104
 data set name levels 124
 default PF key commands 16
 defining PF key functions
 using a BLANK label 73
 using colon (:) to define line commands 72
 Dialog Test
 breakpoint cancel (option C) 279
 changing variable values 295
 DBCS value format 299
 detailed ISPF log entries 311
 dialog trace ISPF log entry 310
 DISPLAY service 313
 hexadecimal format 325
 how dialog parts change 324
 ISPF log entry for current value of variable 312
 modifying a table row 302
 NEWPOOL option 281
 no data written to ISPF log 310
 processing that causes a service call 315
 severe error found at breakpoint 282
 specifying breakpoints 321
 why many ISPF log entries can appear 311
 DISPLAY command 90
 display of data set format 86
 displaying commands after execution 87
 DPLINE process statement 403
 DPLINEC process statement 403
 entering generation data set name 32
 F (free unused space) line command 141
 FIND command 142
 FIND command search strings 91
 Foreground Assembler processing 216
 HEX command 95
 INCLUDE statement 232
 invalid load module directory fields 42
 invalid mixed data 51
 ITEXT data set names 227
 JCL generation for compilers 256
 library concatenation 34
 Linkage Editor concatenation sequence 232
 Linkage Editor search sequence 232
 LNCT process statement 404
 LOCATE command 44, 97, 143, 170

LSTCOLM process statement 405
 member expansion
 code that cannot be processed 212
 member list with expanded line command
 field 140
 member name in title 30
 member parts list 247
 member selection list with statistics 40
 menu 5
 naming a command table 159
 NCHGT process statement 407
 NTITLE process statement 408
 object data set names 218
 OCHGT process statement 407
 panel format 7
 pattern, member selection list 39
 PDF terminal operations 56
 PFSHOW command 73
 PL/I optimizing compiler 268
 print output data set names 243
 printing panels with long lines 77
 printing, deleting, and renaming members 48
 S (select) line command 45
 SAVE command 143
 Search-For list data set name 191
 search-for listing data set name 200
 Search-For process statements 193
 SELECT command 45
 SELECT process statement 409
 selecting a tutorial topic 336
 selecting an option 5
 Session Manager control 206
 SHOWCMD command 144
 SORT command 146, 171
 specifying a qualifier with an asterisk 125
 splitting the display screen 21
 SRCHFOR process statement 412
 SRCHFORC process statement 412
 stacking commands 16, 66
 substituting '/' for data set name 132
 SuperC list data set name 175, 185
 SuperC process statements 195
 SWAP command 22
 symbolic debug data set names 243
 three-level library hierarchy 33
 typing over parameters 27
 typing over scroll amount 27
 uses for library levels 34
 using GROUPS FOR PRIMARY MEMBERS
 field 246
 using SuperC 375
 using the Library utility 48
 value in CHANGED field 40
 value in CREATED field 40
 value in LIB field 40
 value in VV.MM field 40
 VOLUME view display 128
 VS COBOL II allocation data sets
 SYSIN 367
 SYSLIB 367

SYSPRINT 368
 SYSPUNCH 369
 SYSTEM 368
 SYSUT1 369
 SYSUT2 - SYSUT5 369
 VTOC display 136
 with the TSO/E Information Center
 Facility 239
 with TSO/E Information Center Facility 218
 327x terminal keys 19
 EXECUTION PARMS field 208, 242
 Foreground PL/I checkout compiler (option
 4.4) 226
 Exit (option X)
 bypassing the log and list data set panel 23
 ending PDF from the ISPF/PDF Primary Option
 Menu 23
 ending PDF on a split screen 22
 overview 7
 similarities to RETURN command 24
 use of log/list defaults 25
 used with jump function (=) 25
 Exit (option 7.X)
 ending Dialog Test 326
 expansion trigger, defined 211
 EXPIRATION DATE field 110
 EXPIRES field 131
 EXPIRES keyword
 SORT command 146
 explosion chain, defined 249
 extended binary coded decimal interchange code
 Browse or Edit DBCS data as EBCDIC 51
 defining data type 167
 format definition 50
 hexadecimal format 51
 invalid mixed data 51
 MIXED MODE field 50
 using numbered data 168
 Extended Search-For - Concatenation Data Set
 Entry panel 192
 Extended Search-For Compare utility 188
 applications 375
 options
 Blank - search-for strings 191
 E - edit Search-For statements data set 193
 P - select Search-For process options 192
 process options 378
 process statements 393
 return codes 377
 SuperC program description 372
 utility differences 371
 Extended Search-For process options
 ANYC 379
 APNDLST 381
 COBOL 379
 DPACMT 380
 DPADCMT 380
 DPBLKCL 380
 DPCBCMT 380
 DPFTCMT 380

DPPLCMT 380
 DPPSCMT 380
 IDPFX 381
 LMTO 381
 LONGLN 381
 LPSF 382
 LTO 382
 NOPRTCC 382
 NOSEQ 379
 SEQ 379
 Extended Search-For process statements
 * and * (comment lines) 401
 CMPCOLM (compare columns) 396
 CMPLINE (compare line) 399
 DPLINE (do not process line) 402
 DPLINEC (continuation of do not process
 line) 402
 LNCT (line count) 404
 LSTCOLM (list columns) 405
 NCHGT (new change text) 406
 notation conventions 393
 NTITLE (new title) 408
 SELECT (select PDS members) 409
 SLIST (include statements in listing) 411
 SRCHFOR 412
 SRCHFORC 412
 Extended Search-For Utility panel 189

F

F (foreground) parameter
 MODE field 173
 F (free unused space) line command
 Data Set List utility (option 3.4) 141
 F (function) pool
 POOL field 318
 F record format
 RECORD FORMAT field 109
 false match correction notice 356
 FBA record format
 RECORD FORMAT field 76
 FIELD LENGTH field 167
 field mark character 19
 FIELD NUMBER field 167
 FIELD TYPE field 167
 field value abbreviations 365
 field1 parameter
 SORT command 46, 146
 field2 parameter
 SORT command 46, 146
 FILE comparison of a complete PDS 359
 FILE comparison of a sequential data set or
 membered PDS 358
 FILE parameter
 Compare Type field 183
 FIND command
 Browse (option 1)
 character strings, use of 93

 column limitations 94
 conditions for character string matches 92
 displaying column numbers searched 91
 omitting string delimiters 91
 one-byte hexadecimal strings 94
 picture strings, use of 93
 rules for using col-1 and col-2 94
 specifying FIND strings 90
 starting point, direction, and extent of
 search 91
 string not found actions 92
 text strings, use of 92
 using RFIND 94
 using string delimiters 91
 Data Set List utility (option 3.4) 142
 parameters
 CHARS|PREFIX|SUFFIX|WORD 90
 col-1 and col-2 90
 NEXT|ALL|FIRST|LAST|PREV 91
 string 90
 find-mask parameter
 NCHGT process statement 406
 OCHGT process statement 406
 FIRST parameter
 FIND command 91
 FMVLNS process option
 SuperCE utility (option 3.13) 381
 FO parameter
 ATTRIBUTES field 42
 Foreground (option 4) 205
 Assembler H (option 4.1) 219
 Assembler XF (option 4.1A) 220
 COBOL interactive debug (option 4.10A) 241
 compilation limitations 22
 expanding packed data 211
 FORTRAN interactive debug (option 4.11) 244
 input data sets 215
 Linkage Editor (option 4.7) 231
 list data sets 216
 Member Parts List (option 4.12) 245
 object data sets 217
 OS/VS COBOL compiler (option 4.2A) 223
 output listings 54
 overview 6
 Pascal/VS compiler (option 4.6) 229
 password protection 217
 PL/I checkout compiler (option 4.4) 226
 PL/I optimizing compiler (option 4.5) 228
 processing sequence 205
 SCRIPT/VS (option 4.9) 233
 using the TSO/E Information Center
 Facility 218
 VS COBOL II compiler (option 4.2) 221
 VS COBOL II interactive debug (option
 4.10) 240
 VS FORTRAN compiler (option 4.3) 225
 Foreground Assembler H panel 219
 Foreground Assembler XF panel 220
 Foreground Environment Feature 233
 Foreground Linkage Edit panel 231
 Foreground Member Parts List panel 245

Foreground OS/VS COBOL Compile panel 223
 Foreground Pascal/VS Compile panel 229
 Foreground PL/I Checkout Compile panel 226
 Foreground PL/I Optimizing Compile panel 228
 Foreground Print Options for Style panel 236
 with the TSO/E Information Center Facility 239
 Foreground Print Options panel 210
 with TSO/E Information Center Facility 218
 Foreground Selection Panel 205
 Foreground VS COBOL II Compile panel 221
 Foreground VS COBOL II Interactive Debug panel 240
 Foreground VS FORTRAN Compile panel 225
 format definition 50
 Format Definition panel 167
 format definitions 50
 FORMAT NAME field 50, 166
 effect on MIXED MODE field 50
 Format Selection List panel 170
 Format Specification panel 166
 Format Specification utility (option 3.11)
 IBM 5550 terminal support 50
 line commands
 D (deleting a format) 170
 R (renaming a format) 171
 S (selecting a format) 171
 U (updating a format) 171
 options
 A - add a new format 167
 C - copy formats 168
 D - delete a format 169
 L or Blank - display format list 169
 U - update a format 169
 primary commands
 LOCATE 170
 SORT 171
 format, panel 7
 formatted data set 50
 FORTRAN interactive debug (option 4.11) 244
 FORTRAN Interactive Debug panel 244
 FREE DSCBS field 137
 FREE EXTENTS field 137
 FREE parameter
 ACTION field 121
 FREE SPACE field 137
 FROM FORMAT field 166
 FROM TABLE field 167
 FSCR parameter
 MODE field 288, 289
 FUNCTION field 316, 318, 320, 324
 parameter
 Function Traces panel 316
 Functions (option 7.1)

testing a dialog function 287

G

generation data sets 32
 GENERIC UNIT field 108
 GENHELP keyword 164
 GET operation
 OPERATION field 318
 Go (option G) 323
 graphics interface mode, effect on Session Manager 207
 GROUP field 30
 GROUPS FOR PRIMARY MEMBERS field 246, 249
 GWCBL process option
 SuperCE utility (option 3.13) 381

H

HALF parameter
 SCROLL field 8
 Hardcopy utility (option 3.6)
 additional batch printing information 153
 source and index listings 339
 using the TSO/E Information Center Facility 153
 Hardcopy Utility panel
 after JCL generation 152
 before JCL generation 150
 with the TSO/E Information Center Facility installed 154
 hashsum, defined 358
 HELP command
 displaying column numbers searched 91
 ISPF 12, 20
 Tutorial (option T) 338
 HELP reference 165
 Help Tutorial panel 338
 HEX command
 Browse (option 1) 95
 parameters
 DATA 95
 OFF 95
 ON 95
 VERT 95
 hex-offset parameter
 CMPBOFS process statement 395
 hexadecimal data
 Dialog Test variables format 296

I

- I (data set information) line command
 - Data Set List utility (option 3.4) 139
- I (implicit) variable
 - P (pool) field 294
- I (insert) line command
 - Breakpoints (option 7.8) 320
 - Command Table utility (option 3.9) 162
 - Dialog Test (option 7) 285
 - Tables (option 7.4) 303, 305
 - Traces (option 7.7) 317, 318
 - Variables (option 7.3) 294
- IBM Products (option 9) 329
 - options
 - 1 - CSP/AD 330
 - 2 - CSP/AE 331
 - 3 - INFO/SYS 332
 - 4 - COBOL/SF-F 333
 - 5 - COBOL/SF-B 334
 - overview 6
- ID field 41, 289
- IDPFX process option
 - Extended Search-For utility (option 3.13.S) 381
- IEBCOPY utility 103, 117, 141
- IEBGENER utility 153
- IF PARTITIONED, REPLACE LIKE-NAMED MEMBERS field 117
- IF SEQUENTIAL, "TO" DATA SET DISPOSITION field 117
- INCLUDE statement 211
- index listings
 - index listings for load libraries 342
 - index listings for source libraries 341
- index listings for load libraries 342
- index listings for source libraries 341
- INFO/SYS (option 9.3)
 - Information/System 332
- Information Center Facility 150, 157, 209, 235
- Information/System panel 332
- INIT field 41
- INIT keywords 164
- INITIAL DISPLAY VIEW field 125
 - views
 - ATTRIB 126
 - SPACE 126
 - TOTAL 126
 - VOLUME 126
- INITIAL MACRO field 100
- input data sets 215
- INPUT FIELD PAD field 65
- installation exits
 - compress request 103, 141
 - data set allocation 107
 - data set list (DSLIST) 124
 - print utility 150, 157, 209, 235
 - promote 120
- Interactive System Productivity Facility
 - relationships to PDF 1
- internal character representations for APL keyboards 363
- internal character representations for TEXT keyboards 364
- interpretable text data sets 227
- introduction to PDF
 - commands, entering 14
 - entering commands with PF keys 19
 - interacting with PDF 4
 - invoking PDF 4
 - ISPF/PDF primary options 5
 - remembering information with user profiles 28
 - understanding PDF panels 7
 - using the default PF key settings 20
- Invoke Dialog Function/Selection Menu 287
- Invoke Dialog Service panel 313
- IS data set organization
 - DSORG field 131
- ISPCMD5 159
- ISPF
 - See Interactive System Productivity Facility
- ISPF commands
 - assigning 70
 - CURSOR 13
 - DOWN 12, 26
 - END 12, 23
 - ENVIRON 13, 81
 - HELP 12, 20
 - ISPPREP 13
 - jump function (=) 13
 - KEYS 13, 19
 - LEFT 12, 26
 - LIST 13
 - LOG 13
 - PANELID 13
 - PFSHOW 13, 72
 - PRINT 13
 - PRINT-HI 13
 - PRINTG 13, 78
 - PRINTL 13
 - PRINTLHI 13
 - RCHANGE 12, 25
 - RETRIEVE 12
 - RETURN 12, 24
 - RFIND 12, 25
 - RIGHT 12, 26
 - SPLIT 12, 21
 - SPLITV 13, 66
 - SWAP 12, 22
 - taking a screen snapshot 53
 - UP 12, 26
- ISPF ENVIRON Command Settings panel 81
- ISPF library names 29
- ISPF library, defined 29
- ISPF log listings 343
- ISPF member statistics 39
- ISPF Parameter Options panel 63
- ISPF Params (option 0) 63
 - Display (option 0.4) 74
 - Environ (option 0.7) 81
 - Graphic (option 0.6) 78

- List (option 0.5) 76
- Log/List (option 0.2) 67
- overview 6
- PF Keys (option 0.3) 70
- Terminal (option 0.1) 64
- ISPF Tutorial and Introduction panels 336
- ISPF/PDF command notation 15
- ISPF/PDF Primary Option Menu 5
- ISPF/PDF primary options
 - option selection 5
- ISPPREP command
 - ISPF 13
- ISPTLIB 159
- ISRCFIL control file 327
- ISRFORM table 168
- ISRLEMX 256
- ISRSCAN 256
- ISU data set organization
 - DSORG field 131
- ITEXT data sets
 - See interpretable text data sets

J

- JCL
 - See job code language
- JCL generation
 - Assembler and Linkage Editor 257
 - compilers 256
- job code language
 - generating and submitting print jobs 151
- job statement information
 - parameters
 - (ACCOUNT) 56
 - /* lines 56
 - 'NAME' 56
 - user-ida 55
 - preventing JES line counting 236
- JOB STATEMENT INFORMATION field 24, 68
- JOBID field 156
- JOBNAME field 156
- jump function (=)
 - ISPF 13, 25

K

- K (keep) parameter
 - Process option field 24, 67
- K (key) variable
 - T (variable types) field 300, 302, 304
- Katakana terminals 65
- keys
 - ATTENTION (PA1) 18
 - CLEAR 19
 - cursor movement 22

- ERASE INPUT 19
- ISPF 13
- PA1 277
- RESET 19, 277
- RESHOW (PA2) 19
- KEYS command
 - ISPF 19, 70
- KEYS field 306
- keyword parameter
 - CMPBOFS process statement 395
 - CMPLINE process statement 399
- keyword/operand abbreviations 366
- KN (keep and allocate new data set) parameter
 - Process option field 24, 67

L

- LABEL fields 72
- label parameter
 - LOCATE command 97
- LANG field 288
- LANGUAGE field 246
- LARGEST field 137
- LAST DATE MODIFIED field 307, 309
- LAST MODIFIED BY field 308, 309
- LAST parameter
 - FIND command 91
- LAST SERVICE RETURN CODE field 309
- LAST TABLE SERVICE field 309
- LAST TIME MODIFIED field 307, 309
- lcmd parameter
 - SELECT command 45
- LEFT command
 - ISPF 12, 26
- LEL
 - See Linkage Editor language
- LIB field 40, 42
- libraries and data sets
 - data set passwords 49
 - format definitions 50
 - list and log data sets 52
 - member selection lists 37
 - mixed mode 50
 - naming ISPF libraries and data sets 29
 - packed data sets 51
 - sample PDF session 56
 - using LMF to control ISPF libraries 32
- library 33
- library concatenation
 - during editing 35
 - during language processing 35
 - functions that use concatenation 35
 - GROUP field 34
- library controls 32
- Library Management Utilities panel 327
- library member modification and promotion
 - locking members while editing 36
- Library utility (option 3.1)

member list line commands 47
 options
 B - browse member 103
 Blank - display member list 102
 C - compress data set 103
 D - delete member 104
 I - data set information 104
 L - print entire data set 104
 P - print member 103
 R - rename member 103
 S - data set information (short) 105
 X - print index listing 103
 source and index listings 339
 Library Utility panel 102
 Line Cmd field 237
 line commands
 Command Table utility (option 3.9) 162
 Data Set List utility (option 3.4) 131, 138
 defined 6, 14
 Dialog Test (option 7) 285
 Format Specification utility (option 3.11) 171
 member selection list 43
 SCRIPT/VS (option 4.9) 237
 LINE LENGTH field 77
 line-number parameter
 LOCATE command 97
 LINE parameter
 Compare Type field 183
 MODE field 288, 289
 Lines per page field 67
 Linkage Editor (option 4.7) 231
 concatenation sequence 232
 LEL control statements 232
 Linkage Editor (option 5.7) 270
 Linkage Editor concatenation sequence 232
 Linkage Editor language 232
 LINKAGE EDITOR OPTIONS field
 Foreground Linkage Editor (option 4.7) 231
 List (option 0.5)
 printing panels with long lines 77
 list and log data sets
 foreground and batch output listings
 list-id parameter 54
 prefix parameter 54
 userid parameter 54
 JOB STATEMENT INFORMATION 55
 list data set 53
 log data set 53
 other temporary data sets 54
 prefix parameter 52
 printing list/log data sets 53
 taking a screen snapshot 53
 temporary names 52
 userid parameter 52
 LIST command
 ISPF 13
 List Data Set Characteristics panel 76
 list data sets 216
 parameters
 list-id 44, 143
 prefix 44, 143
 LIST ID field 208, 216, 253
 list-id parameter
 Foreground and Batch output listings 54
 SAVE command 44, 143
 LIST keyword 164
 list of abbreviations 365
 command names 365
 field values 365
 keywords/operands 366
 programming languages 366
 scroll amounts 366
 List VTOC utility (option 3.7), removal of 155
 LISTING DS NAME field 175, 191
 Listing Dsn field 184
 LISTING DSNNAME field 200
 listing formats
 data set list listings 347
 ISPF log listings 343
 member list listings 344
 source and index listings 339
 SuperC listings 348
 LISTING TYPE field 175, 184
 LM Utilities (option 8) 327
 overview 6
 LMF migration utility 287
 LMTO process option
 Extended Search-For utility (option 3.13.S) 381
 LNCT process statement
 Extended Search-For utility (option 3.13.S) 404
 parameter
 nnnnnn 404
 SuperCE utility (option 3.13) 404
 load module library
 invalid directory fields (?) 42
 member statistics 41
 load module library display 41
 expanded line command field 141
 Local printer ID field 24, 67
 LOCATE command
 Breakpoints (option 7.8) 320
 Browse (option 1) 96
 Data Set List utility (option 3.4) 143
 Dialog Test (option 7) 284
 Format Specification utility (option 3.11) 170
 member selection list 43
 parameters
 label 97
 line-number 97
 lparm 143
 name 170
 string 43, 285
 Tables (option 7.4) 301, 303, 305, 306
 Traces (option 7.7) 317, 318
 Variables (option 7.3) 294
 LOCK field 36
 locking members while editing
 conditions when locking causes errors 37
 conditions when locking is ignored 37
 locking members with copy/lock 119
 LOCS process option
 SuperCE utility (option 3.13) 381

- Log (option 7.5)
 - displaying the ISPF log 310
 - logged trace output 310
 - reasons log is not available 310
- log and list data set default options
 - changing primary and secondary page values 69
- Log and List Defaults panel 67
- LOG command
 - ISPF 13
- log data set 53
- Log/List (option 0.2)
 - changing defaults 7
 - effect on Exit (X) option 25
 - log and list data set default options 68
- LOGICAL RECORD LENGTH field 76
- LONG MESSAGE field 8
- LONG parameter
 - LISTING TYPE field 175, 184
- LONGLN process option
 - Extended Search-For utility (option 3.13.S) 381
 - SuperCE utility (option 3.13) 381
- lparm parameter
 - LOCATE command 143
- LPSF process option
 - Extended Search-For utility (option 3.13.S) 382
- LRECL field 131
- LRECL keyword
 - SORT command 146
- LSTCOLM process statement
 - Extended Search-For utility (option 3.13.S) 405
 - parameters
 - start-colm 405
 - stop-colm 405
 - SuperCE utility (option 3.13) 405
- LTO process option
 - Extended Search-For utility (option 3.13.S) 382

M

- M (display member list) line command
 - Data Set List utility (option 3.4) 139
- M (machine) parameter
 - PRINTER CARRIAGE CONTROL field 156
- M (mixed) parameter
 - FIELD TYPE field 167
- M record format
 - RECORD FORMAT field 109
- MAX parameter
 - SCROLL field 8
- member expansion
 - block size 212
 - COPY statement 214
 - FIELD A 214
 - FIELD B 214
 - IDENTIFICATION DIVISION 214
 - language processing table 212
 - restrictions 213
 - return codes 213

- trigger statement errors 215
- WITH DEBUGGING MODE clause 214
- MEMBER field 30
- member list display 40
 - expanded line command field 140
- member list listings
 - load libraries 345
 - source libraries 344
- member list lists 344
- member not found 249
- member parameter
 - BROWSE command 88
 - SELECT process statement 409
 - SYSTEM DD statement 257
- Member Parts List (option 4.12) 245
 - CALL statement format 245
 - member not found 249
 - options
 - 1 - browse/print member parts list 247
 - 2 - write member parts data set 248
- PROCEDURE DIVISION 245
 - record format 248
- Member Parts List (option 5.12) 273
- member parts list display 247
- member selection lists
 - displaying member lists 38
 - fields 40
 - load module library 42
 - source library 40
- ISPF member statistics 39
- line commands
 - B (browse) 47
 - D (delete) 47
 - E (edit) 47
 - P (print) 47
 - R (rename) 47
 - S (select) 45
- list data set 44
- load module library member statistics 41
- primary commands
 - LOCATE 43
 - RESET 44
 - SAVE 44
 - SELECT 45
 - SORT 46
- saving scroll value 27
- table of differences 38
- updating a member list 49
- membered PDS, defined 174
- menus 9
- Message Display Panel 292
- MESSAGE field 129
- MESSAGE ID field 291
- MESSAGE keyword
 - SORT command 146
- messages
 - as means of communication 4
 - conditions for display 16
 - displayed by HELP command 20
 - failure to enter required value 9

- inconsistent values 9
- LONG MESSAGE field 8
- SHORT MESSAGE field 8
- mixed data
 - assumed 51
 - B (browse data set) line command 138
 - defining data type 167
 - E (edit data set) line command 138
 - format definition 50
 - invalid, examples of 51
 - MIXED MODE field 50
 - non-mixed mode 51
 - unformatted 51
 - using numbered data 168
- MIXED MODE field 50
- MOD field 41
- MOD parameter
 - IF SEQUENTIAL, "TO" DATA SET
 - DISPOSITION field 117
- MODE field 173, 288, 289
- MODIFIED ROW COUNT field 308, 309
- Modify Row panel 302
- MOUNT authority 32, 108
- move members 35
- MOVE parameter
 - ACTION field 121
- Move/Copy utility (option 3.3)
 - alias entries 123
 - options
 - C and CP - copying data sets 116
 - L and LP - copying and locking data sets 119
 - M and MP - moving data sets 118
 - P and PP - promoting data sets 120
 - source and index listings 339
 - with load modules 122
 - without load modules 122
- Move/Copy Utility panel 115
- multiple search strings 200
- Multiple Search Strings panel 201
- MULTIPLE STRINGS field 200

N

- N (name) variable
 - T (variable types) field 300, 302, 304
- N (non-modifiable) attribute
 - A (variable attributes) field 294
- NAME field 40, 42, 129
- NAME keyword
 - LOCATE command 170
 - SORT command 146, 171
- NAME parameter
 - JOB STATEMENT INFORMATION 56
- NAMES field 306
- naming ISPF libraries and data sets
 - ISPF library names 29

- OTHER PARTITIONED OR SEQUENTIAL
 - DATA SET NAMES 31
 - volume serials 32
- NARROW process option
 - SuperCE utility (option 3.13) 382
- national language support ix
- native mode 7
- NBTM keyword
 - CMPBOFS process statement 395
 - CMPLINE process statement 399
- NCHGT process statement
 - Extended Search-For utility (option 3.13.S) 406
 - parameters
 - 'find-mask' 406
 - 'string' 406
 - SuperCE utility (option 3.13) 406
- NEVER parameter
 - LOCK field 36
- New DS Name field 182
- new-member parameter
 - SELECT process statement 409
- NEW OUTPUT CLASS field 156
- new-title-text parameter
 - NTITLE process statement 408
 - OTITLE process statement 408
- NEW USERID field 147
- NEW VERSION NUMBER field 147
- NEWAPPL field 289
- NEWNAME field 102
- NEWPOOL field 289
- NEXT parameter
 - FIND command 91
- nnnnn parameter
 - LNCT process statement 404
- NO parameter
 - ACTIVE field 316, 318, 320
 - BROWSE Output field 186
 - CONFIRM DELETE REQUEST field 127
 - IF PARTITIONED, REPLACE LIKE-NAMED MEMBERS field 117
 - LOCK field 36
 - MIXED MODE field 51
 - MULTIPLE STRINGS field 200
 - NEWAPPL field 289
 - NEWPOOL field 289
 - PACK DATA field 122
 - PASSLIB field 290
 - PURGE DATA SET field 113
 - SPECIFY PACK OPTION FOR "TO" DATA SET field 117
 - TABLE AVAILABLE field 307
 - TABLE ON DISK field 309
- NOCC parameter
 - DISPLAY command 89
- NOLIST parameter
 - LISTING TYPE field 175, 184
- non-paired change, defined 356
- NOP action
 - ACTION column 161
- NOPRTCC process option
 - Extended Search-For utility (option 3.13.S) 382

SuperCE utility (option 3.13) 382
 NOSEQ parameter
 SEQUENCE NUMBERS field 176
 NOSEQ process option
 Extended Search-For utility (option 3.13.S) 379
 SuperCE utility (option 3.13) 379
 NOT OPEN parameter
 STATUS FOR THIS SCREEN field 307, 308
 NOWRITE parameter
 OPEN OPTION field 309
 NTITLE process statement
 Extended Search-For utility (option 3.13.S) 408
 parameter
 new-title-text 408
 SuperCE utility (option 3.13) 408
 NTOP keyword
 CMPBOFS process statement 395
 CMPLINE process statement 399
 NUMBER OF COPIES field 154
 NUMBER OF KEYS field 306
 NUMBER OF NAMES field 306
 NUMBER OF PF KEYS field 65
 NUMBER OF ROWS field 306
 NX parameter
 ATTRIBUTES field 42



object data sets 217
 OBJECT field
 Batch PL/I checkout compiler (option 5.4) 267
 Foreground PL/I checkout compiler (option 4.4) 226
 Foreground VS FORTRAN compiler (option 4.3) 225
 OBTM keyword
 CMPBOFS process statement 395
 CMPLINE process statement 399
 OCHGT process statement
 SuperCE utility (option 3.13) 406
 OFF parameter
 CONFIRM command 142
 HEX command 95
 SHOWCMD command 144
 OL parameter
 ATTRIBUTES field 42
 Old DS Name field 182
 old-member parameter
 SELECT process statement 409
 OLD parameter
 IF SEQUENTIAL, "TO" DATA SET
 DISPOSITION field 117
 ON parameter
 CONFIRM command 142
 HEX command 95
 SHOWCMD command 144
 OPEN OPTION field 309
 OPEN parameter

STATUS FOR THIS SCREEN field 307, 308
 operand notation
 one required ({ }) 15
 optional ([]) 15
 OR symbol (|) 15
 OPERATION field 318
 OPT field 288
 OPTION field 8
 entering commands 14
 option selection 5
 C - Changes 7
 T - Tutorial 7
 X - Exit 7
 0 - ISPF Parms 6
 1 - Browse 6
 2 - Edit 6
 3 - Utilities 6
 4 - Foreground 6
 5 - Batch 6
 6 - Command 6
 7 - Dialog Test 6
 8 - LM Utilities 6
 9 - IBM Products 6
 ORIGINAL ROW COUNT field 308, 309
 OS/VS COBOL compiler (option 4.2A) 223
 OS/VS COBOL compiler (option 5.2A) 264
 OTHER field
 Batch Assembler H (option 5.1) 258
 Batch Assembler XF (option 5.1A) 260
 Batch Linkage Editor (option 5.7) 270
 Batch OS/VS COBOL compiler (option 5.2A) 264
 Batch PL/I checkout compiler (option 5.4) 267
 Batch VS COBOL II compiler (option 5.2) 262
 Batch VS FORTRAN compiler (option 5.3) 265
 Foreground OS/VS COBOL compiler (option 4.2A) 223
 Foreground Pascal/VS compiler (option 4.6) 229
 Foreground PL/I checkout compiler (option 4.4) 226
 Foreground PL/I optimizing compiler (option 4.5) 228
 Foreground VS COBOL II compiler (option 4.2) 221
 Foreground VS FORTRAN compiler (option 4.3) 225
 OTHER PARTITIONED OR SEQUENTIAL DATA
 SET NAMES 31
 other temporary data sets 54
 OTITLE process statement
 SuperCE utility (option 3.13) 408
 OTOP keyword
 CMPBOFS process statement 395
 CMPLINE process statement 399
 OUTDD DD field 178
 Outlist utility (option 3.8)
 options
 Blank - display job output 158
 D - delete job output from SYSOUT hold
 queue 157

- L - list job names/IDs via the TSO STATUS command 157
- P - print job output and delete from SYSOUT hold queue 157
- R - requeue job output to a new output class 158
- source and index listings 339
- Outlist Utility panel 156
- OV parameter
 - ATTRIBUTES field 42
- OVSUM listing type
 - FILE comparison of a complete PDS 359
 - FILE comparison of a sequential data set or membered PDS 358
- OVSUM parameter
 - LISTING TYPE field 175, 184

P

- P (pool) field 294
- P (prefix) qualifier
 - SRCHFOR process statement 412
- P (print) line command
 - Data Set List utility (option 3.4) 47, 139
 - Library utility (option 3.1) 47
- P (profile) pool
 - POOL field 318
- P (profile) variable
 - P (pool) field 294
- PACK DATA field 121
- packed data sets 51
 - requirements for using 52
- packing data
 - Foreground (option 4) 206
- PAGE parameter
 - SCROLL field 8
- paired change, defined 356
- PANEL field 287
- panel format 7
- PANEL NAME field 291
- panel types 8
 - data entry panels 9
 - menus 9
 - scrollable data displays 9
- PANELID command
 - ISPF 13
- panels
 - as means of communication 4
 - basic types 8
 - data set list display view panels
 - ATTRIB view panel 129
 - SPACE view panel 128
 - TOTAL view panel 129
 - VOLUME view panel 128
 - default mode 7
 - defined 7
 - load module library display 41

- member list display 40
 - expanded line command field 140
- menus 9
 - code, blank 9
 - code, defined 9
- native mode 7
- panel format 7
- retaining previous values 9
- scrollable format 9
- split screen example 21
- what they display 4
- Panels (option 7.2)
 - testing dialog panels 291
- PARM field 289
- Pascal/VS compiler (option 4.6) 229
- Pascal/VS compiler (option 5.6) 269
- PASSLIB field 290
- PASSTHRU action
 - ACTION column 161
- PASSWORD field 208, 217
- password protection 217
- pattern parameter
 - SELECT command 45
- pattern, defined 39
- PA1 (ATTENTION) key 18, 277
- PA2 (RESHOW) key 19
- PCF
 - See Programming Control Facility
- PD (print and delete) parameter
 - Process option field 24, 67
- PDF
 - See Program Development Facility
- PDS Member List field 183
 - parameters 189
- PF Key Definitions and Labels - Alternate Keys
 - panel 71
- PF Key Definitions and Labels - Primary Keys
 - panel 71
- PF Key Definitions and Labels panel 70
- PF Keys (option 0.3)
 - assigning ISPF/PDF commands 19, 70
 - assigning labels 72
 - informing ISPF of the number of PF keys 65
 - terminals with 12 PF keys 70
 - terminals with 24 PF keys 70
- PF keys, default assignments 11
- PFSHOW command
 - ISPF 13, 72
- PGM field 289
- picture strings, use of 93
- PL/I checkout compiler (option 4.4) 226
- PL/I checkout compiler (option 5.4) 267
- PL/I optimizing compiler (option 4.5) 228
- PL/I optimizing compiler (option 5.5) 268
- PO data set organization
 - DSORG field 131
- POOL field 318
- POU data set organization
 - DSORG field 131
- prefix parameter
 - FIND command 90

- Foreground and Batch output listings 54
- list and log data sets 52
- member list list data set 44, 143
- SYSTEM DD statement 257
- PREV parameter
 - FIND command 91
- primary commands
 - Browse (option 1) 87
 - Command Table utility (option 3.9) 162
 - Data Set List utility (option 3.4) 142
 - defined 6, 14
 - Format Specification utility (option 3.11) 170
 - member selection list 43
- primary member, defined 249
- Primary pages field 68
 - changing values 69
- PRIMARY QUANTITY field 108
- PRINT command
 - ISPF 13
- Print Graphics Parameters panel 78
- PRINT-HI command
 - ISPF 13
- PRINT ID field 242, 243
- PRINT MODE field 236
- print output data sets 243
- print utility exit 150, 157, 209, 235
- PRINTER CARRIAGE CONTROL field 156
- PRINTER FORMAT field 154
- PRINTER LOCATION field 153
- PRINTG command
 - ISPF 13, 78
- printing
 - list/log data sets 53
 - panels with long lines 77
- printing a SuperCE listing in batch mode 188
- printing your output 150
- PRINTL command
 - ISPF 13
- PRINTLHI command
 - ISPF 13
- PROCEDURE DIVISION 245
- Process option field 24, 67
- Process Options field 185, 191
- PROFILE DSN field 172
- PROFILE NAME field 100
- profiles
 - user 28
- PROG ID field 242
- program access (PA) keys 18
- Program Development Facility
 - interacting with 4
 - introduction to 1
 - invoking 4
- program function (PF) keys 19
- Programming Control Facility 49
- programming language abbreviations 366
- project administrator 327
- PROJECT field 29
- promote exits 120
- promote members 35, 36

- DSORG field 131
- PSU data set organization
 - DSORG field 131
- PURGE DATA SET field 113
- PUT operation
 - OPERATION field 318

Q

- QUAL command
 - Breakpoints (option 7.8) 320
 - Dialog Test (option 7) 285
- QUAL field 320
- QUALIFICATION PARAMETER VALUES
 - field 321
- qualifier parameter
 - SRCHFOR process statement 412

R

- R (rename) line command
 - Data Set List utility (option 3.4) 47, 139
 - Format Specification utility (option 3.11) 171
 - Library utility (option 3.1) 47
- R (repeat) line command
 - Breakpoints (option 7.8) 320
 - Command Table utility (option 3.9) 162
 - Dialog Test (option 7) 286
 - Tables (option 7.4) 303, 305
 - Traces (option 7.7) 317, 318
- RACF
 - See Resource Access Control Facility
- RCHANGE command
 - ISPF 12, 25
- REASON CODE field 121
- RECFM field 131
- RECFM keyword
 - SORT command 146
- RECORD FORMAT field 76, 109
- RECORD LENGTH field 109
- REFERRED field 131
- REFERRED keyword
 - SORT command 146
- REFMOVR process option
 - SuperCE utility (option 3.13) 382
- Rename Data Set panel 112
- repeating a FIND or CHANGE command
 - using the RCHANGE command 25
 - using the RFIND command 25
- RESET command
 - Browse (option 1) 97
 - member selection list 44
- Reset ISPF Statistics panel 147
- Reset ISPF Statistics utility (option 3.5)
 - options

- D - delete ISPF statistics 149
- R - reset (create/update) ISPF statistics 148
- results of resetting statistics 149
- valid logical record lengths 147
- RESET key 19, 277
- RESET MOD LEVEL field 147
- RESET SEQ NUMBERS field 148
- RESHOW (PA2) key 19
- Resource Access Control Facility 49
- restrictions on member expansion and member parts lists
 - all programming languages 213
 - Assembler 213
 - COBOL 214
 - FORTRAN 214
 - Pascal 214
 - PL/I 214
 - SCRIPT/VS 214
- RESUME command
 - Dialog Test (option 7) 285
- RETRIEVE command
 - ISPF 12
- retrieving commands 27
- RETURN CODE field 325
- return codes
 - Batch JCL generation 256
 - member expansion 215
 - SuperC 377
- RETURN command
 - ISPF 12, 24
- RETURN keyword 164
- RF parameter
 - ATTRIBUTES field 42
- RFIND command
 - Data Set List utility (option 3.4) 142
 - ISPF 12, 25
- RIGHT command
 - ISPF 12, 26
- RN parameter
 - ATTRIBUTES field 42
- Rnn parameter
 - WHEN field 320
- ROW IDENTIFICATION field 298
- ROW NUMBER field 301
- RU parameter
 - ATTRIBUTES field 42
- rules for:
 - printing panels with long lines 77
 - service call image for function trace entries 311
 - specifying log and list options 68
 - substituting '/' for data set name 132
 - typing over multiple-choice parameters 27
 - typing over scroll amounts 27
 - using the add row option 305
 - using the col-1 and col-2 operands 94
 - using the Dialog Test Breakpoints option 322
 - using the Modify Row option 302

S

- S (information (short)) line command
 - Data Set List utility (option 3.4) 139
- S (save) variable
 - T (variable types) field 300, 302, 304
- S (select) line command
 - Format Specification utility (option 3.11) 171
 - member selection list 45
 - SCRIPT/VS (option 4.9) 237
- S (shared) pool
 - POOL field 318
- S (shared) variable
 - P (pool) field 294
- S (suffix) qualifier
 - SRCHFOR process statement 412
- S record format
 - RECORD FORMAT field 109
- sample PDF session 56
- SAVE command
 - Data Set List utility (option 3.4) 143
 - member list listings, examples of 344
 - member selection list 44
 - parameters
 - list-id 44, 143
- SC parameter
 - ATTRIBUTES field 42
- Screen Containing Graphics to be Printed 79
- SCREEN FORMAT field 66
- SCRIPT/VS (option 4.9) 233
 - changing style options 238
 - line commands
 - D (delete) 237
 - S (select) 237
 - processing sequence 233
 - selecting a formatting style 237
 - using the TSO/E Information Center Facility 239
- SCRIPT/VS Options for Style panel 238
- SCRIPT/VS Processor panel 233
- scroll amount abbreviations 366
- scroll amounts 26
- scroll commands 26
- SCROLL field 8
- scrollable data displays 9
 - data set format 10
 - table format 9
- scrolling within a data set
 - typing over scroll and multiple-choice parameters 26
 - using the scroll commands 26
- Search DS Name field 189
- Search-For Process Statements panel 193
- Search-For utility (option 3.14)
 - applications 375
 - return codes 377
 - selecting members from a member list 202
 - SuperC program description 372
 - utility differences 371

Search-For Utility panel 199
 Search Member List panel 203
 SEARCH STRING field 199
 search strings, multiple 200
 Secondary pages field 68
 changing values 69
 SECONDARY QUANTITY field 109
 SELECT action
 ACTION column 160
 SELECT command
 Format Specification utility (option 3.11) 171
 member selection list 45
 parameters
 SELECT process statement
 Extended Search-For utility (option 3.13.S) 409
 parameters
 member 409
 new-member 409
 old-member 409
 SuperCE utility (option 3.13) 409
 Select SCRIPT/VS Formatting Style panel 237
 SELECT service 163, 280
 SEQ parameter
 SEQUENCE NUMBERS field 176
 SEQ process option
 Extended Search-For utility (option 3.13.S) 379
 SuperCE utility (option 3.13) 379
 SEQUENCE NUMBERS field 176
 SERVICE field 319
 Service Name field 324
 Session Manager
 Command (option 6) 277
 Foreground (option 4) 206
 Member Parts List (option 4.12) 246
 SCRIPT/VS 235
 Session Manager, using 277
 SETVERB action
 ACTION column 161
 shift-in characters 50
 DBCS character strings 51
 Dialog Test variables 297
 formatted data Edit and Browse 50
 invalid mixed data, examples of 51
 treatment in non-mixed mode 51
 using the VALUE field 299
 shift-out characters 50
 DBCS character strings 51
 Dialog Test variables 297
 formatted data Edit and Browse 50
 invalid mixed data, examples of 51
 treatment in non-mixed mode 51
 using the VALUE field 299
 SHORT MESSAGE field 8
 SHOWCMD command
 Data Set List utility (option 3.4) 144
 parameters
 OFF 144
 ON 144
 SHR NOWRITE parameter
 OPEN OPTION field 309
 SHR WRITE parameter
 OPEN OPTION field 309
 SI characters
 See shift-in characters
 single-byte character set
 See extended binary coded decimal interchange code
 SIZE field 40, 42, 137
 SLIST process statement
 Extended Search-For utility (option 3.13.S) 411
 SuperCE utility (option 3.13) 411
 snapshot of a screen 53
 SO characters
 See shift-out characters
 SORT command
 Data Set List utility (option 3.4) 146
 Format Specification utility (option 3.11) 171
 keywords
 %USED 146
 BLKSZ 146
 CREATED 146
 DEVICE 146
 DSORG 146
 EXPIRES 146
 LRECL 146
 MESSAGE 146
 NAME 146
 RECFM 146
 REFERRED 146
 TRACKS 146
 VOLUME 146
 XT 146
 member selection list 46
 parameters
 field1 46, 146
 field2 46, 146
 NAME 171
 TIME 171
 source and index listings
 index listings 341
 source listings 340
 source data for SuperC listings 349
 SOURCE DATA ONLINE field 252
 SOURCE DATA PACKED field 206, 211, 252
 SOURCE field 242
 source listings 340
 source segments 35
 SOURCE TYPE field 208
 SPACE display view
 INITIAL DISPLAY VIEW field 126
 SPACE UNITS field 108
 Special Display Panel 314
 Specify Disposition of Log and List Data Sets
 panel 23
 SPECIFY PACK OPTION FOR "TO" DATA SET
 field 117
 SPF
 See System Productivity Facility
 SPFUTIL program 163
 SPLIT command 21
 ISPF 12, 21

- repositioning a split screen 21
- split screen limitations 21
- splitting and swapping display screens
 - using the SPLIT command 21
 - using the SWAP command 22
- SPLITV command
 - ISPF 13, 66
- SRCHFOR process option
 - Extended Search-For utility (option 3.13.S) 412
 - parameters
 - qualifier 412
 - string 412
 - qualifiers
 - blank 412
 - P (prefix) 412
 - S (suffix) 412
 - W (word) 412
- SRCHFORC process option
 - Extended Search-For utility (option 3.13.S) 412
- stacking commands, defined 16
- standard Search-For utility
 - entering multiple search strings 200
- start-colm parameter
 - CMPCOLM process statement 396
 - CMPCOLMN process statement 397
 - CMPCOLMO process statement 397
 - CMPLINE process statement 400
 - DPLINE process statement 402
 - DPLINEC process statement 402
 - LSTCOLM process statement 405
 - NCHGT process statement 407
 - OCHGT process statement 407
- START COLUMN field 167
- Statements Dsn field 185, 191
- statistics
 - ISPF members 39
 - load module libraries 41
- statistics, creating or resetting 148
- Status for Table panel
 - table not open 307
 - table open 308
- STATUS FOR THIS SCREEN field 307, 308
- stop-colm parameter
 - CMPCOLM process statement 396
 - CMPCOLMN process statement 397
 - CMPCOLMO process statement 397
 - CMPLINE process statement 400
 - DPLINE process statement 402
 - DPLINEC process statement 402
 - LSTCOLM process statement 405
 - NCHGT process statement 407
 - OCHGT process statement 407
- string parameter
 - DPLINE process statement 402
 - DPLINEC process statement 402
 - FIND command 90
 - LOCATE command 43, 285
 - NCHGT process statement 406
 - OCHGT process statement 406
 - SRCHFOR process statement 412
 - STYLE field 383, 438
- SUFFIX parameter
 - FIND command 90
- SuperC Activate/Create Profile options
 - A - activate 197
 - C - create 197
- SuperC compare types
 - BYTE 184
 - FILE 183
 - LINE 183
 - WORD 183
- SuperC listing source data 349
- SuperC listing types
 - CHNG 175, 184
 - DELTA 175, 184
 - LONG 175, 184
 - NOLIST 175, 184
 - OVSUM 175, 184
- SuperC Member List panel 177
- SuperC process options
 - ANYC 379
 - APNDLST 381
 - APNDUPD 383
 - CNPML 381
 - COBOL 379
 - DLMDUP 381
 - DLREFM 381
 - DPACMT 380
 - DPADCMT 380
 - DPBLKCL 380
 - DPCBCMT 380
 - DPFTCMT 380
 - DPPLCMT 380
 - DPPSCMT 380
 - FMVLNS 381
 - GWCBL 381
 - LOCS 381
 - LONGLN 381
 - NARROW 382
 - NOPRTCC 382
 - NOSEQ 379
 - REFMOVR 382
 - SEQ 379
 - UPDCMS8 383
 - UPDCNTL 384
 - UPDMVS8 387
 - UPDPDEL 388
 - UPDSEQ0 389
 - UPDSUMO 389
 - WIDE 382
 - XWDCMP 382
- SuperC Process Statements panel 195
- SuperC Utility - Submit Batch Jobs panel 178
- SuperC Utility - Submit Batch Jobs panel using
 - NOLIST 181
- SuperC utility (option 3.12)
 - "new" data set 172
 - "old" data set 174
 - applications 375
 - printing a SuperC listing in batch mode 177
 - process options 378
 - process statements 393

- return codes 377
- specifying members on a member list 176
- SuperC program description 372
- SuperC submit options 179
 - blank - generate output listing to SYSOUT CLASS 179
 - 1 - generate output listing in DATA SET NAME 179
 - 2 - generate output listing using completed //OUTDD DD 180
- utility differences 371
- SuperC Utility panel 172
- SuperCE - Concatenation Batch Entry panel 188
- SuperCE - Concatenation Interactive Entry panel 187
- SuperCE - Profile Manager panel 196
- SuperCE process statements
 - * and .* (comment lines) 401
 - CMPBOFS (compare byte offsets) 395
 - CMPCOLM (compare columns) 396
 - CMPCOLMN (compare column new) 397
 - CMPCOLMO (compare column old) 397
 - CMPLINE (compare line) 399
 - DPLINE (do not process line) 402
 - DPLINEC (continuation of do not process line) 402
 - LNCT (line count) 404
 - LSTCOLM (list columns) 405
 - NCHGT (new change text) 406
 - notation conventions 393
 - NTITLE (new title) 408
 - OCHGT (old change text) 406
 - OTITLE (old title) 408
 - SELECT (select PDS members) 409
 - SLIST (include statements in listing) 411
- SuperCE Profile Browse Display panel 196
- SuperCE utility (option 3.13)
 - applications 375
 - options
 - A - activate/create profiles 195
 - B - submit batch data set compare 187
 - Blank - compare data sets 186
 - E - edit SuperCE statements data set 194
 - P - select process options 378
 - P - select SuperCE process options 194
 - S - Extended Search-For Compare utility 188
- process options 378
- process statements 393
- return codes 377
- SuperC program description 372
- SuperCE utility 182
- utility differences 371
- SuperCE Utility panel 182
- SWAP command
 - ISPF 12, 22
- symbolic debug data sets 242
- synchronization data 373
- SYSIN data set
 - Assembler H 367
 - VS COBOL II 367

- Assembler H 367
- VS COBOL II 367
- SYSLIN data set
 - Assembler H 368
 - VS COBOL II 368
- SYSOUT class field 24, 67, 178, 253
- SYSPRINT data set
 - Assembler H 368
 - VS COBOL II 368
- SYSPUNCH data set
 - Assembler H 369
 - VS COBOL II 369
- system command table 159
- System Productivity Facility 163
- SYSTEM data set
 - Assembler H 368
 - VS COBOL II 368
- SYSTEM DD statement
 - parameters
 - member 257
 - prefix 257
- SYSUT1 data set
 - Assembler H 369
 - VS COBOL II 369
- SYSUT2 - SYSUT5 data sets
 - VS COBOL II 369

T

- T (truncated) attribute
 - A (variable attributes) field 294, 300, 302
- T (truncation) column 160
- T (variable types) field 300, 302, 304
- T record format
 - RECORD FORMAT field 109
- TABLE AVAILABLE field 307
- table format 9
- table input library 159
- TABLE NAME field 298, 301
- Table of Contents panel 337
- TABLE ON DISK field 309
- Table Structure panel 306
- Tables (option 7.4)
 - Add Row line commands
 - D (delete) 305
 - I (insert) 305
 - R (repeat) 305
 - Add Row primary commands
 - CANCEL 305
 - END 305
 - LOCATE 305
 - Display Row commands
 - END 301
 - LOCATE 301
 - Display Structure command
 - LOCATE 306
 - Modify Row line commands
 - D (delete) 303

- I (insert) 303
- R (repeat) 303
- Modify Row primary commands
 - CANCEL 303
 - END 303
 - LOCATE 303
- options
 - 1 - display row 300
 - 2 - delete row 301
 - 3 - modify row 301
 - 4 - add row 304
 - 5 - display structure 306
 - 6 - display status 307
- working with dialog tables 298
- Tables panel 298
- Target Panel for Promoting 121
- temporary data sets 54
- TERM field
- TERMIN command procedure statements 277
- Terminal (option 0.1)
 - changing default delimiter 16
 - informing ISPF of the number of PF keys 65
 - overriding mode switching 7
- Terminal Characteristics panel 64
- TERMINAL TYPE field 64
- terminals
 - Batch Assembler H (option 5.1) 258
 - Batch Assembler XF (option 5.1A) 260
 - Batch Linkage Editor (option 5.7) 270
 - Batch VS COBOL II compiler (option 5.2) 262
 - Batch VS FORTRAN compiler (option 5.3) 265
 - 12 PF keys 70
 - 24 PF keys 70
- TEST command
 - TSO 279
- TEST field
 - Batch OS/VS COBOL compiler (option 5.2A) 264
 - Foreground OS/VS COBOL compiler (option 4.2A) 223
 - Foreground VS COBOL II compiler (option 4.2) 221
- testing dialogs 279
- text strings, use of 92
- TIME CREATED field 307, 309
- TIME parameter
 - SORT command 171
- TITLE field 8
- To Panel for Copying 116
- To Panel for Moving 118
- TOP parameter
 - BY ROW NUMBER field 298
- TOTAL display view
 - INITIAL DISPLAY VIEW field 126
- trace output in ISPF log
 - function trace entries 311
 - trace header entries 310
 - variable trace entries 311
- Traces (option 7.7)
 - Function Traces line commands
 - D (delete) 317

- I (insert) 317
- R (repeat) 317
- Function Traces primary commands
 - CANCEL 317
 - END 317
 - LOCATE 317
- options
 - 1 - function traces 315
 - 2 - variable traces 317
- Variable Traces line commands
 - D (delete) 318
 - I (insert) 318
 - R (repeat) 318
- Variable Traces primary commands
 - CANCEL 318
 - END 318
 - LOCATE 318
- working with trace specifications 315
- Traces panel 315
- TRACKS field 130, 137
- TRACKS keyword
 - SORT command 146
- transmission codes 363
- trigger statement errors 215
- TRKS parameter
 - SPACE UNITS field 108
- TRKS/CYLS field 137
- TS parameter
 - ATTRIBUTES field 42
- TSO Command Processor panel 275
- TSO commands 22
 - ACCOUNT 32, 108
 - ALLOCATE 107
 - COPY 159
 - DELETE 127, 128, 142
 - TEST 279
- TSO commands and CLISTs
 - Data Set List utility (option 3.4) 132
- variables
 - Data Set List utility (option 3.4) 134
 - ZDLBLKSZ 134
 - ZDLCAT 135
 - ZDLCDATE 134
 - ZDLCONF 134
 - ZDLDEV 134
 - ZDLDSN 134
 - ZDLDSORG 134
 - ZDLEDAT 134
 - ZDLEXT 134
 - ZDLLCMD 134
 - ZDLLRECL 134
 - ZDLMSG 135
 - ZDLNDSN 135
 - ZDLRDATE 134
 - ZDLRECFM 134
 - ZDLREF 135
 - ZDLsize 134
 - ZDLUSED 134
 - ZDLVOL 134
- TSO commands and CLISTs, entering 276
- TTR data 103

TTR field 42
 Tutorial (option T) 335
 finding ISPF information 4
 HELP command 338
 selecting 335
 selecting the tutorial
 ending the Tutorial 337
 selecting a tutorial topic 335
 Tutorial (option 7.T)
 displaying tutorial information 326
 TYPE field 30
 typing over scroll and multiple-choice
 parameters 26

U

U (uncatalog data set) line command
 Data Set List utility (option 3.4) 139
 U (updating a format) line command
 Format Specification utility (option 3.11) 171
 U record format
 RECORD FORMAT field 109
 understanding PDF panels 7
 panel format 7
 panel types 8
 UNIT field 137
 unmovable data sets 131
 UP command
 ISPF 12, 26
 UPD parameter
 BROWSE Output field 186
 UPDATE COUNT field 308, 309
 UPDATE DS NAME field 174
 Update Dsn field 185
 updating a member list 49
 UPDCMS8 process option
 SuperCE utility (option 3.13) 383
 UPDCMS8 update data set 383
 UPDCNTL process option
 SuperCE utility (option 3.13) 384
 UPDCNTL update data sets and formats 384
 UPDMVS8 process option
 SuperCE utility (option 3.13) 387
 UPDPDEL process option
 SuperCE utility (option 3.13) 388
 UPDPDEL update data set 388
 UPDSEQ0 process option
 SuperCE utility (option 3.13) 389
 UPDSEQ0 update data set 389
 UPDSUMO process option
 SuperCE utility (option 3.13) 389
 UPDSUMO update data set 389
 user-ida parameter
 JOB STATEMENT INFORMATION 55
 user profiles 28
 userid parameter

Utilities (option 3) 101
 Command Table utility (option 3.9) 159
 Convert Menus/Messages utility (option
 3.10) 163
 Data Set List utility (option 3.4) 124
 Data Set utility (option 3.2) 106
 Format Specification utility (option 3.11) 166
 Hardcopy utility (option 3.6) 150
 Library utility (option 3.1) 102
 List VTOC utility (option 3.7), removal of 155
 Move/Copy utility (option 3.3) 115
 Outlist utility (option 3.8) 156
 overview 6
 Reset ISPF Statistics utility (option 3.5) 147
 Search-For utility (option 3.14) 199
 SuperC utility (option 3.12) 172
 SuperCE utility (option 3.13) 182
 Utility Selection Menu 101

V

V (VDEFINE) variable
 P (pool) field 294
 V record format
 RECORD FORMAT field 109
 VALUE field 294, 299, 300, 302, 304
 specifying a DBCS value 299
 VARIABLE field 293, 300, 302, 304, 317
 Variable Traces panel 317
 variable usage, Dialog Test 281
 Variables (option 7.3)
 line commands
 D (delete) 294
 I (insert) 294
 manipulating variables
 creating new variables 295
 deleting variables 295
 primary commands
 CANCEL 294
 END 294
 LOCATE 294
 usage notes
 DBCS data 297
 hexadecimal data 296
 input errors 295
 split-screen mode 296
 test mode 296
 variable life 296
 variable value 296
 working with dialog variables 293
 Variables panel 293
 VBA record format
 RECORD FORMAT field 76
 VERB column 160
 VERT parameter
 HEX command 95
 VIO, allocating temporary data sets 54
 VOLUME DATA field 137

VOLUME display view
 INITIAL DISPLAY VIEW field 126
 VOLUME field 125, 130
 VOLUME keyword
 SORT command 146
 VOLUME SERIAL field 32, 108
 VS COBOL II compiler (option 4.2) 221
 allocation data sets
 SYSIN 367
 SYSLIB 367
 SYSLIN 368
 SYSPRINT 368
 SYSPUNCH 369
 SYSTERM 368
 SYSUT1 369
 SYSUT2 - SYSUT5 369
 VS COBOL II compiler (option 5.2) 262
 VS COBOL II interactive debug (option 4.10) 240
 VS COBOL II interactive debug (option 5.10) 272
 VS COBOL II Interactive Debug panel 272
 VS data set organization
 DSORG field 131
 VS FORTRAN compiler (option 4.3) 225
 VS FORTRAN compiler (option 5.3) 265
 VTOC DATA field 137
 VTOC information display 136
 VV.MM field 40

W

W (word) qualifier
 SRCHFOR process statement 412
 WHEN field 320
 WIDE process option
 SuperCE utility (option 3.13) 382
 window as scrolling concept 25
 WORD parameter
 Compare Type field 183
 FIND command 90
 WRITE parameter
 OPEN OPTION field 309
 writing a list to a list data set
 data set list 143
 member list 44

X

X (print index listing) line command
 Data Set List utility (option 3.4) 139
 XT field 130
 XT keyword
 SORT command 146

XWDCMP process option
 SuperCE utility (option 3.13) 382

Y

YES parameter
 ACTIVE field 316, 318, 320
 BROWSE Output field 186
 CONFIRM DELETE REQUEST field 127
 IF PARTITIONED, REPLACE LIKE-NAMED MEMBERS field 117
 LOCK field 36
 MIXED MODE field 51
 MULTIPLE STRINGS field 200
 NEWAPPL field 289
 NEWPOOL field 289
 PACK DATA field 122
 PASSLIB field 290
 PURGE DATA SET field 113
 SPECIFY PACK OPTION FOR "TO" DATA SET field 117
 TABLE AVAILABLE field 307
 TABLE ON DISK field 309
 YYYY.DDD or YY.DDD format
 EXPIRATION DATE field 111
 YYYY/MM/DD or YY/MM/DD format
 EXPIRATION DATE field 111

Z

Z (compress data set) line command
 Data Set List utility (option 3.4) 141
 ZCMD input field 164
 ZDLBLKSZ variable 134
 ZDLCAT variable 135
 ZDLCDATE variable 134
 ZDLCONF variable 134
 ZDLDEV variable 134
 ZDLDSN variable 134
 ZDLDSORG variable 134
 ZDLEDATE variable 134
 ZDLEXT variable 134
 ZDLLCMD variable 134
 ZDLLRECL variable 134
 ZDLMSG variable 135
 ZDLNDSN variable 135
 ZDLRDATE variable 134
 ZDLRECFM variable 134
 ZDLREF variable 135
 ZDLsize variable 134
 ZDLUSED variable 134
 ZDLVOL variable 134

Numerics

1 to 9999 parameter

 SCROLL field 8

3277KN parameter

 TERMINAL TYPE field 65

3278CF parameter

 TERMINAL TYPE field 65

3278KN parameter

 TERMINAL TYPE field 65

3850 virtual volume, allocating a data set to 108

3850 virtual volumes, accessing 32

5550 terminal 166

60 lines per page

 Lines per page field 67

80 lines per page

 Lines per page field 67

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. It will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Possible topics for comments are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name and mailing address:

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

Cut or Fold Along Line

What is your occupation? _____

NW: 15457 DocId: 70002077 Page 833
Number of latest Technology Newsletter (if any) concerning this publication: _____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

Cut or Fold Along Line

Reader's Comment Form

Fold and tape

Please Do Not Staple

Fold and tape



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

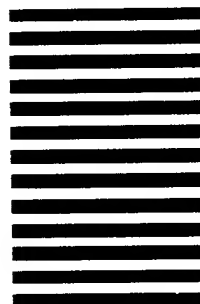
FIRST CLASS

PERMIT NO. 40

ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

**INTERNATIONAL BUSINESS MACHINES CORPORATION
DEPARTMENT T45 INFORMATION DEVELOPMENT
PO BOX 60000
CARY NC 27511**



Fold and tape

Please Do Not Staple

Fold and tape

